

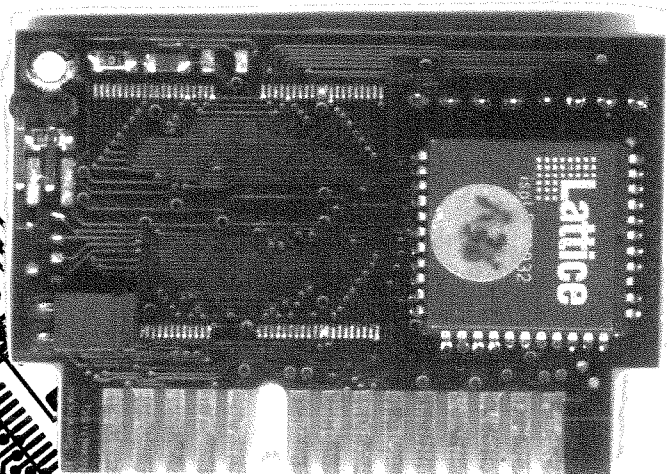
QL Today

Volume 2
Issue 6
March/April
1998

ISSN 1432-5454

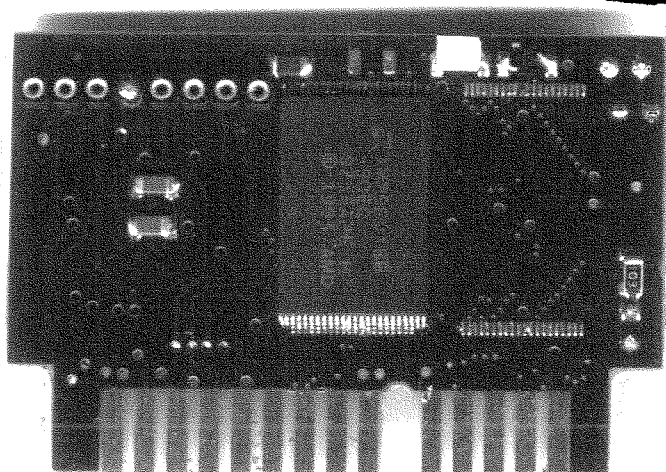
The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...

STUART HONEYBALL
EXPLAINS IN DETAIL
HOW IT ALL WORKS



WITH
CIRCUIT
DIAGRAM

THE ROMDISQ



Now
AVAILABLE!

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software Tel. +49 203 502011
Im stillen Winkel 12 Fax +49 203 502012
47169 Duisburg Box1 +49 203 502013
Germany Box2 +49 203 502014

English office:

Q Branch Tel. +44 1273 386030
P.O. Box 7 Fax +44 1273 381577
East Sussex, United Kingdom BN41 2ND

Editor:

Dilwyn Jones Tel. +44 1248 354023
41 Bro Emrys Fax +44 1248 354023
Tal-Y-Bont, Bangor
Gwynedd, United Kingdom LL57 3YT

QL Today is published bi-monthly, our volume begins on 15th of May. Subscriptions begin with the current issue at the time of sign up. Subscription rates are as follows:

Germany	DM 70,-
England	DM 60,- or £25
Rest of the World	DM 70,- or £30

Payment in DM (drawn on a German bank) can be made by either Cheque or Euro-Cheque. Payment in £ (drawn on an English bank) can be made by Cheque. Cheques should be made payable to Jochen Merz Software (German office) or QBranch (English office).

Credit Card holders may subscribe by either calling or sending their Credit Card number and Expiry date. Credit cards will be charged in DM (German office) or in £ (English office).

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) in ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising DEADLINES are as follows:

Issue 1: 15 April	Issue 2: 15 June
Issue 3: 15 August	Issue 4: 15 October
Issue 5: 15 December	Issue 6: 15 February

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine is copyrighted and all material published remains the property of **QL Today** unless otherwise specified. Written permission from **QL Today** is required before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

Contents

- 3 Editorial - Dilwyn Jones
- 3 Editorial 2 - Jochen Merz
- 4 News
- 8 Thierry's Great Web Site - Tim Swenson
- 9 QRoute UK - Road Journey Planner for Qdos & SMSQ/E - Simon N. Goodwin
- 17 Netted - Roy Wood
- 24 QXL matters(?) - H. Huyg
- 26 RWAP QL Software - who are they??? - Rich Mellor
- 27 Glossary of Abbreviations and Terms: S ... Z - Dilwyn Jones
- 28 The use and abuse of floppy disks and drives - Part 2 - Mark Knight
- 32 Beginners Club - Andrea and Paolo Carpi
- 33 Logical Fundamentals - Part 4 - The ROMDISQ Logic - Stuart Honeyball
- 39 The future of ProWesS - Part 1 - Joachim Van der Auwera
- 42 Byts of Wood 2 - Roy Wood
- 46 Gee Graphics ! (On the QL ?) - Part 3 - H. L. Schaaf
- 50 Letter-Box
- 51 Filter Programs - Norman Dunbar
- 55 QL Show USA



Advertisers

in alphabetical order

Jochen Merz Software	7
PROGS - Van der Auwera	19
QBox USA	37
QBranch	40, 41
Quanta	47
Qubbesoft	45
RWAP Software	15
W.N. Richardson (EEC)	29
TF Services	23

Editorial

Dilwyn Jones

It's that time again, when the QL Today team set about copying all those floppy disks for the latest cover disk. We've included copies of some PD/Freeware software reviewed or mentioned in this magazine in the last year, such as Jonathan Hudson's acclaimed Qascade program launcher for pointer environment users. The cover disk will have a file on it called README.TXT (plain text version) which you should read before trying to use the disk. Jochen and I will keep our fingers crossed that the zip files are useable this time....

The software scene is starting to look up recently, what with two new pointer driven word-processors in preparation, one called Proverb from Wolfgang Lernerz, the other called Paragraph, by Francois Lanciault in Canada. Both should be capable of utilising the powers of Prowess, including scaleable fonts and possibly inclusion of graphics within documents, as found on modern word processors on other computers. QL Today will try to keep you informed of progress! And speaking of Prowess, Joachim van der Auwera of PROGS has sent us an article about the future of Prowess, reflecting both his commitment to Prowess and to the future of the QL in general.

The RomDisq project from TF Services got delayed at the final stages of production, which unfortunately lead to customers' credit cards being debited well in advance of receiving their goods. While this is regrettable, Tony Firshman did at least write to customers informing them of the situation and explaining what had gone wrong. Tony also answered questions via the QL Users Email Mailing List, keeping people up to date of the situation. At the time of writing, some units had been sent to customers with the final problems being sorted out for Gold Card users.

Stuart Honeyball writes in this issue about the hardware side of the RomDisq, so that you can see what makes it tick, and this follows on nicely from Stuart's recent hardware articles. I hope Stuart's current project, a replacement for the ZX8301 giving better QL displays actually gets to market, unlike some of his recent ideas!

Dilwyn Jones

Editorial 2

Dear QL-Today Readers,

we are looking back at two volumes of QL Today. Two excellent volumes, as many readers tell us. The result shows that we like doing QL Today. However, without you, the authors, QL Today would not be possible. I would like to thank all the authors for their contributions. I would also like to thank Dilwyn for his work ... without him, QL Today would not be possible either. Finally, thanks to all the readers ... we know there are lots of people out there who appreciate our work.

If you dislike something in QL Today, please let us know. If you like QL Today, tell us and others!

We mentioned this in a previous issue: it is not always possible to include every article/review in the next upcoming issue. Sometimes, we're in the lucky situation of having more material than a single issue can hold. We do our best to create an issue which contains a large variety of material from as many authors as possible, trying to consider how urgently an article needs to be set. Therefore, if a contribution does not appear in the next issue, please be patient. If you don't find it in the next issue, please contact us - something may have happened.

We're only human, doing a lot of other things to earn money, therefore errors may happen. Fortunately, not many serious errors have happened so far. Last issue created an unhappy situation: due to lots of work, illness and communication problems both at Dilwyn's and my side, Simon N Goodwin's QRoute review did not make it into issue 5. We figured it all out after the issue was done - and squeezing it in would have forced us to throw out a number of other articles - the issue contents would have lost their balance. I apologize to Simon for having to hold it back while he thought it was urgent, but at the time the decision had to be made it was the only possible solution. My feeling is that most kinds of articles can be split across issues if too long, but reviews should not be split.

I don't want to go into detail of all the decisions we have to make, to ask around for articles if it's time for the next issue and we're short of articles ... and sometimes get an overwhelming amount of feedback. Good! Great! This is what keeps us happily going to create a magazine for YOU, with YOUR help!

Looking forward towards another interesting volume of QL Today,

Jochen Merz

News

News from QUBBESoft P/D

Ron Dunnett reports that he has recently discovered that the Seagate ST31277 Ultra-DMA hard drive cannot successfully be used with the Qubide interface. The drive goes to sleep and cannot be woken up by Qubide. Ron suggests that users wishing to purchase a hard disk drive for use with Qubide avoid this drive for now, or at least to check with him at Qubbesoft P/D to see if there has been any progress on compatibility.

New Version of FILEINFO II

(Source: QL Users Mailing List)

Thierry Godefroy has announced the release of version 3.10 of the popular extension for QPAC2, File Info. This release offers the following enhancements:

- The 'S*BASIC commands' action type now also works when job 0 is asleep (in a button)
- The File Info II 'view file' thread is now the child of job 0 so that it does not disappear if the calling job terminates.
- The action selection menu now has a selection keystroke for each item.
- The 'Fileinfo II history' thing has been implemented and related Fileinfo II extensions and S*BASIC procedures/functions have been added
- The new 'Fileinfo II thread' thing allows you to call Fileinfo II as an executable thing - it also supports events and history handling.
- The 'Fileinfo2_bin' file is now configurable using Config or MenuConfig, you can configure the 'system asleep' button name, the history parameters, the menu titles and colours, and the menu items selection method.

Fileinfo II v3.10 can be downloaded from Thierry's WWW site at:

<http://www.imagnet.fr/~godefroy/english/download.html>

News from QBRANCH

We have made preliminary contact with Peter Graf with regard to marketing the Q40 when it is available.

We are looking into the prospect of building a SCSI interface for the QL/Aurora. There is not much news on this yet but we will be looking

for volunteers to write the software drivers to connect to various devices that people may want to connect to it. This is not a trivial task so we need all the help that we can get to bring this off.

Mark Knight tells me that there may soon be a new version of The Knight Safe featuring compression of the Archive onto the backup medium and de-compression on restoring.

We are offering free upgrades to Geoff Wicks Thesaurus program v1.00 to v1.01 (this is a bug fix only) send us the master disk and return postage.

We may soon be selling the new Pointer Driven version of Solvit2 plus and offering upgrades to existing owners (ring for details). I have a Beta test version which looks good and I am awaiting a new release version.

Another new release will be Rich Mellor's new book. This is the SBASIC/SuperBASIC Reference Manual. A complete list of all of the keywords in all of the toolkits from QDOS through to SMSQ/E giving details of the various differences between QDOS/Minerva/SMSQ usage and lots of programming tips. It will come with three disks, holding all the toolkits and an electronic index. Phone for price details from mid March onwards.

Last and definitely not least we are joining forces with TF Services to produce a box to put your Aurora system into. There will be a limited number of these systems available (working name 'The Minis-QL'). It will have a laptop sized footprint but will be about twice as tall. There will be room in the box for an Aurora, Super Gold Card, Qubide, 3.5" floppy and 2.5" Hard disk. There will also be a power supply and a Ni-Cad battery which will give about an hour's usage off the mains. TF Services are looking into an LCD screen for this as an extra unit but the standard one will give the usual Aurora output. Several things need to be resolved before this is ready for release but the prototype is built and running and will be on show at Hove.

QBranch now has its own Web site, currently consisting of 6 areas, QBranch contact details, QDOS/SMSQ software information, future developments, QL Today, Hardware and a section devoted to Roy Wood's sound engineering activities. For those who are on the net, you can find the site at

<http://www.qbranch.demon.co.uk>

QL Today paid a visit to this site and found that an example of a forthcoming product from QBranch could be a music program for the QL, something we are lacking at the moment.

News from PROGS

I have found a comment on the FSF web site that the LZW patent does not cover decompression. Therefore there is now a GIF picture driver part of ProWesS. Or you can get it on my web site (news page). This means it is now possible to display gif files in the ProWesS reader.

There is also an update of PWfile. Some interesting improvements

- filenames can now be stuffed when indicated
- the filetype is now displayed for files which are not "normal". This is displayed instead of the version. An abbreviation is used for some known uses of filetypes. If the filetype is not recognized, the number is displayed, preceded by a 't'. The abbreviations are: **exe** for executables, **rel** for relocatable files, **xmod** for external moduled (syslib filetype), **TPs4** for The PAINTER mode 4 compressed screen, **TPs8** for The PAINTER mode 8 compressed screen, **TPp** for The PAINTER pattern, **TPf** for The PAINTER font.
- PWfile can now execute an executable without FileInfo II (when using the FileInfo II link). This way, you don't have to reconfigure FileInfo II and this will work even if FileInfo II is not loaded. This new behaviour is configurable.
- when you copy files to MSDOS media (with any of the copying variants), PWfile will now automatically replace the extension separator by a dot and truncates the filename to 8+3 format as expected on these media. (Please note that you need a version of syslib_next with dates from 17.11.1997 or later for this detection to work).

And some small bugfixes. The current version is v1.02.

QL Hackers Journal News

Following a period of non-appearance due to what publisher Tim Swenson described as feeling 'burnt out quite a bit', QL Hackers Journal, the electronic fan-zine for the QL, has now reappeared. Issue 27 was published in January 1998 and hopefully will now continue to appear on a regular basis.

Issues of QL Hackers Journal are available electronically via email or via the Anon-FTP server, garbo.uwasaf.i, or on disk direct from Tim Swenson at the cost of sending it to you. Back issues are available from many bulletin boards systems or from PD libraries. The e-zine is freely distributable to all QL users.

Issue 27, the most recent at the time of writing, includes articles about Tim's planned book-

on-disk about QLiberator (for which he invites contributions such as hints and tips from QLiberator users), Regular Expressions (e.g. the * and ? wildcard characters) and the GREP package for pattern matching, End Of File checking, Background Programs, MicroEmacs line numbering, and adding Config blocks to BASIC programs via the Basconfig utility.

In additions to the provisionally entitled QLiberator Sourcebook (working title only), Tim invites contributions from readers to the QL Hackers Journal itself.

QL Hacker's Journal, c/o Tim Swenson

38725 Lexington St. #230

Fremont, CA 94536, U.S.A.

Email: swensontc@geocities.com

http://www.geocities.com/SilconValley/Pines/5865/

Club QL International

Another disk-based magazine for the QL, Club QL International's e-zine is now available via email or on disk, and the Club even has its own Web site now, from where some back issues are available. Like the Hackers Journal, Club QL's disk based monthly newsletter is also freely distributable to all QL users. The club is currently looking at finding a librarian to hold back issues of the newsletter on disk for the benefit of those without a modem to download copies electronically. Given that by the time you read this, their 108th issue will have been published, this is quite a lot of reading matter for the QL!

Copies of some back issues can be downloaded from Club QL's Web site at

http://www.geocities.com/siliconvalley/vista/4807

Mike Kenneally, 6 Barnaby Road,

Poynton, Cheshire, SK12 1LR, UK

Telephone: 01625-878207 Fax: 01625-260072

Email: MIKEKENNEALLY1@compuserve.com

JMS News

There are quite a lot of updates on various Jochen Merz Software products. I won't list the minor changes here in detail, you can check the version numbers in the JMS advert: we have improved QSpread, QD, FiFi, Wined and the Thing Extension.

However, recent changes on SMSQ/E are definitely worth mentioning: we discovered that it is impossible to access sectors greater than 1 Gigabyte on ATARI ST, STE or TT's ACSI port. The limit is defined by the ACSI port, which allows only 6-Byte group 0 SCSI commands. The problem here is, that group 0 commands have 21 bits for the sector number. 2 to the power of 21 is 2 Mega of sectors (each 512

bytes), resulting in 1 Gigabyte. The real SCSI-port of the TT also allows group 1 SCSI commands which can handle much larger drives. With SMSQ/E 2.86 or before, 6 byte commands were used here too, which means that the sector numbers "wrap" above 1 Gigabyte. Accessing sector 2²¹ will access sector 0, sector 2²¹+1 will access sector 1. From SMSQ/E 2.87, this is fixed!

From SMSQ/E 2.87 on, it is also possible to save executable programs (e.g. Quill, QD, Clock) to DOS-type disks. They can now be executed from there and remain file type 1, i.e. show the "E" flag in QPAC 2!

Speaking of QPAC 2: in the past, file lists with more than 8192 entries could lead to problems. This is fixed in version 1.39. Fileinfo 2 is now supported, no need to patch QPAC 2 anymore.

PROVERB NEWS

Proverb will be a new wordprocessor running under Prowess, and thus make use of the display possibilities thereof: scaleable fonts, true WYSIWYG (What You See Is What You Get), several columns, inclusion of pictures and (hopefully) tables.

Of course, headers, footers and all the usual stuff will be in there as well.

At the time of writing (mid February), a pre-alpha version of Proverb exists and has gone out to several testers. The program is far from finished and it is not likely that a beta version will be out before April.

The current pre-alpha version allows you to:

- have different typefaces (fonts) and styles (bold, underline etc)
- select blocks of text with the mouse, and apply a certain style to it (e.g. underline)
- print the text
- set margins etc with the mouse

There is still much that remains to be done...

Wolfgang Lenerz

RWAP QL SOFTWARE NEWS

D-DAY MKII, SToQL and Quizmaster II have now been altered to make it easier to run them from hard-disk and sub-directories. Quizmaster II will also now work in both MODE 4 and MODE 8.

A public domain demonstration version of D-DAY MKII is now also available from me or Qubbesoft which incorporates all of the features of the full version except that you must control a fixed small army of Allies against the computer controlled Axis army.

War In the East MKII has never been released commercially (until now). It is a wargame based on the conflict between the German and Rus-

sian Armies during World War II - you play the Germans and the aim is to defeat the Russians, taking the main cities in the former USSR, and hold them against the onslaught of the Russian troops and the Russian winter. Play the computer or another player.

This is unfortunately only currently available as an upgrade to the original War In the East which was limited to playing against the computer and extremely slow. The computer intelligence has now been improved to such an extent that it is very difficult to beat the computer through all four scenarios.

War In the East MKII forms a useful complement to D-DAY MKII.

Open Golf v5.17 has just been released, which fixes a bug which meant the program crashed when used under SMSQ/E.

Q-Route v1.04 has just been released which implements a few minor bug fixes and also incorporates faster loading and map drawing times. Q-Route is available from QBranch.

Rich Mellor

UQLX NEWS from Richard Zidlicky

The QL emulator UQLX (for Unix/Linux etc based machines) can now handle pretty big screens - I have tested up 8192x4096.

Further information is available on this Web site:

http://www.geocities.com/SiliconValley/Bay/2602/uqlx_main.html

QL Today contributor Jonathan Hudson, a regular user of UQLX, writes: "Richard Zidlicky's 021298 release of uqlx supports variable screen sizes (from 512x256, it appears QDOS won't go any smaller) to very large; the current maximum tested is 8192x4096. Limited by memory, common sense and presumably, the inherent limits of 16bit integers."

Jonathan Hudson News

New versions of QTPI (1.63) and QFAX (2.83) are around; there is no real new functionality; they just consolidate some earlier bug fix releases.

QL JOKE BOOK

QL Today contributor Darren Branagh has released what he calls 'the first QL Joke Book'. Basically, this is a compilation of text files containing a huge number of jokes under a variety of subject headings, fronted by a well known Text File Viewer program. You can read about light bulbs, Irish jokes, funny answering machine messages, medical jokes, lawyer jokes and so on.

JOSHEM MARZ SOFTWARE

Im stillen Winkel 12 • 47169 Duisburg • Germany

☎ 0203-502011 Fax 0203-502012 Mailbox 0203-502013 & 502014)

EMAIL: smsq@j-m-s.com http://www.j-m-s.com/smsq/

SMSQ/E V2.87

SMSQ/E is the new operating system which allows you to run your QL programs and adds an enormous amount of additional features: faster, flexible disk format, multiple and much faster BASICs, faster screen driver and much more! New in V2.87: executable files can be saved to DOS disks, Aurora can handle >512 lines, SCSI disks >1GB are supported and more. The current manual is Revision 8 (costs DM 18,-).

SMSQ/E for QXL & QXL 2 DM 199,-
SMSQ/E for ATARIs with QL-Emulator DM 199,-
SMSQ/E for ATARIs without QL-Emulator DM 249,-
SMSQ/E for GoldCard & SuperGoldCard DM 199,-

QPC with SMSQ/E allows you to run most QL software on modern PC's (486 or Pentium, with MSDOS 6 or Windows95). You gain all the benefits of SMSQ/E (which is built into QPC and included in the price!). If you already own SMSQ/E for another system listed above, then you pay only **DM 199,-**. If you do not own SMSQ/E for a different system, then you get QPC with SMSQ/E for only **DM 249,-**.

For **DM 40,-** extra, you get it with CueShell inbuilt!

Special Offers

SyQuest 270MB Removeable Drive
SCSI (used) DM 199,00
Medium for SyQuest 270MB (used) each DM 55,00
Janus-Card (fast ATARI-Emulator for PC) with
68020 (used) DM 299,00
Miracle Expanderam 512k (used) DM 49,00
Toolkit 2 (original-Catridge) (used) DM 25,00
Vortex 386 for ATARI Mega STE
(just the Board) (used) DM 59,00
24 MHz-Accelerator for ATARI Mega ST
(price includes installation!) DM 149,00
4MB memory expansion for ATARI Mega ST
(price includes installation!) DM 129,00
QL-Emulator Mode 4+8 for ATARI Mega ST
(price includes installation!) DM 129,00
Microdrive Cartridge (formats >220 sectors)
(used) each DM 1,50
DD-Disk 720k VERBATIM
available in larger quantities each DM 0,69

Software, Games, Applications & QL Spares

QL Games

BlackKnight Chess ... DM 119,90
 Pipes DM 29,90
 BrainSmasher DM 39,90
 Arcanoid DM 39,90
 Firebirds DM 39,90
 QShang DM 39,90
 Diamonds DM 39,90
 The Oracle DM 39,90
 MineField DM 39,90
 Double Block DM 39,90
 The Lonely Joker 2 ... DM 59,00
 SuperGamesPack ... DM 90,00

QL Spares

ZX8301 DM 19,90
 ZX8302 DM 9,90
 Keyboard membrane .. DM 25,00

An ideal time to get things NOW - the £ is worth DM 3,- (!!!) at the time I write this!

QL Applications

QD Editor [V9.15] . DM 125,00
 QD Upgrade from V8 DM 24,90
 FiFi II File Finder [V4.16] . DM 49,90
 FiFi II Upgrade from previous Version DM 19,90
 QMAKE [V4.21] . DM 44,90
 QLiberator SuperBASIC Compiler . DM 139,00
 QLoad-Ref DM 49,90
 QLQ [V1.13] . DM 69,90
 QMAC Macro Assembler [V1.01] . DM 69,00
 QMENU [V7.04] . DM 41,90
 QMENU Upgrade DM 16,90
 QPAC 1 [V1.05] . DM 61,50
 QPAC 2 [V1.39] . DM 119,00
 QTYP 2 Spell-Checker ... [V2.17] . DM 82,50
 QPTR Pointer Toolkit ... [V0.30] . DM 89,90
 QSpread Spreadsheet ... [V1.43] . DM 169,00
 QSpread Update from V1.34 or before . DM 16,00
 QSpread Update from V1.35-1.41 free
 QSUP [V3.09] . DM 79,90
 EPROM Manager [V3.02] . DM 61,50
 WINED [V1.21] . DM 49,90
 I/O 2 Toolkit [V2.16] . DM 99,00
 BASIC Linker [V1.12] . DM 49,90

Applications

LDUMP [V1.05] ... DM 65,00
 DISA Interactive Disass. . [V3.02] . DM 95,00
 DISA V3 - Upgrade from V1 or V2 . . DM 35,00
 EasyPTR BASIC DM 130,00
 EasyPTR CAsm DM 260,00
 Stylus-Driver for text87 and text91 . DM 69,00
 HyperHelp for BASIC DM 44,90
 DiskMate 5 DM 69,00
 CueShell DM 95,00
 SERMouse Driver DM 29,00
 QDOS/SMS Reference Manual DM 84,90
 Update sheets from March 1997 . . . DM 13,00
 Update sheets from Nov. 1997 DM 13,00

ProWesS + Applications

(all ProWesS Applications require ProWesS which is not included!)
 ProWesS WindowManager+HTML Reader . DM 129,00
 DataDesign Database DM 79,00
 LineDesign Vektor/DesktopPublishing . . . DM 79,00
 fsearch DM 49,00
 Pflist DM 49,00
 fontutils DM 79,00
 PWfile DM 64,00

TERMS OF PAYMENT

Postage and package [Germany] DM 8,99 (if total value of goods is up to DM 50,- then only DM 5,99). [Europe] DM 14,50 (if total value of goods is up to DM 50,- then only DM 9,50). [Overseas] between DM 14,50 (1 item) and DM 35,- (maximum). All



prices incl. 15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.



If you're looking for some humour to brighten up 1998, Darren says the QL Joke Book will be available from the usual sources of QL PD software. Darren asks that if you enjoy it, you consider making a small donation to charity.

Darren D. Branagh, The Falconry, Glenmacnass, Glendalough, Co. Wicklow, Ireland.

QemuLator News

At the time of writing (early March) the Alpha 5 version of the QL-emulator for Windows95/Windows NT, called QemuLator, is now available for download from the Web site:

<http://www.geocities.com/SiliconValley/Heights/1296/winql.html>

Improved features include a speed gain over previous releases, the ability to emulate a QL with up to 4MB of RAM, support for QL disk repair programs, PAR device implemented, improved error reporting and memory access controls, and changes to the configuration system.

Thierry's Great Web Site

Tim Swenson

I've been asked by our Editor to occasionally report on what QL resources and general goings-on are happening on the Internet. As an Internet user for 9 years, I have found that using the Internet to support the QL has brought the QL community closer together, even given how far apart we really are.

Of all of the QL web sites, one has come to stand out amongst all the others, "The Sinclair QL and QDOS Compatible Systems Site" maintained by Thierry Godefroy of France. Thierry is the BBS (Bulletin Board System) Sysop for the QL Contact France (QLCF) BBS and has extended his information sharing duties to the Internet.

When I started my web page about 2 years ago, I tried to have the definitive list of Sinclair web sites and Sinclair e-mail addresses (this included ZX81, Spectrum, T/S 2068, QL, and Z88). The amount of effort to keep this current makes it difficult to keep it current. For QL users, Thierry has put forth the effort to keep, what I feel, is THE definitive listing of QL web sites and QL e-mail addresses. No other QL site has as many links to other QL sites. When I want to find another QL site, it is to Thierry's page I go. Actually, since I don't have Thierry's page bookmarked in Netscape, I go to Yahoo first and find Thierry's page there.

On his page is the following:

Other QL Web Sites: These listings are broken down into the following categories:

Generalist, Hardware and Software Suppliers, Software Developers, Clubs, BBS's, and Publications, Software Downloading, Historical, and Miscellaneous. Thierry goes to a lot of effort to keep his lists current.

QL FTP Sites: These are Anon-FTP servers that hold some QL software.

QL Newsgroups: This lists all of the newsgroups that cover Sinclair computers and the QL and even has links to Gopher access to the Mousnet newsgroups. Before I got local access to the Mousnet newsgroups, I used the link on Thierry's page to read these groups.

Latest News: If Thierry sees something of importance to QLers, he posts it in this section of his web pages.

List of Wired QLers: This is THE list of e-mail addresses of QL users. If you are new to the Internet and want to know what QL users are available via e-mail, this is the list to look at. Most well known developers, vendors, and general QLers have an e-mail address.

Address Book: This is a listing of the Snail mail addresses of QL vendors, groups, and publications.

List of QL Dedicated BBSs: This is a list of QL BBSs all over the world.

Searches for QL sites on the Internet: This takes you to one of the major Internet search engines, with QL already plugged in and searched on.

QLCF BBS: This lists the various file areas on the QLCF BBS and the individual files in these areas.

Sinclair QL Local Newsgroup: This is a newsgroup that Thierry has implemented on his system using Java. It even has an interface for Non-Java web browsers. This section has some problems and I have had difficulty posting to it. There seems to be a bug in the Java code or on the hosting server.

Download Page: This page lists QL software available on Thierry's site and from other sites around the world. If you are looking for a specific Freeware package and can't find it here, ask Thierry if he has it on his BBS and if he can make it available on the web site. For me, Thierry has been very good about this and I thank him greatly, as he is my main source for QL PD/Freeware packages.

QRoute UK

Road Journey Planner for Qdos & SMSQ/E

Simon N. Goodwin

Part-time traffic broadcaster Simon N Goodwin checks out a new release from QBranch which automatically finds short and speedy routes across the British road network.

QRoute is a route planner for the Pointer Environment. Route planners scan a digitised road map, trying to find a route between nominated places, perhaps avoiding others. They know the difference between types of road and let you assign your own average speeds to each category, in miles or kilometres per hour. QRoute has seven categories: motorway/dual carriageway, single carriageways in good and poor condition, single track roads, ferries, and two urban classes, for single and dual carriageway roads.

Dilwyn had no trouble choosing me to review this one, as since the demise of QL World I've eeked out part of my living as a traffic broadcaster, freelancing for the Midlands Roadwatch service which supplies traffic and travel information to BBC local radio and a handful of independent stations in a 13-county area. This makes me well aware of the pitfalls of automatic route planning: abnormal loads, roadworks, accidents and congestion are the indigestible staple diet at Roadwatch.

When it comes to avoiding congestion, QRoute is no substitute for an RDS radio - and ideally a time machine, as even the speediest traffic broadcasters can rarely warn you about unexpected snarl-ups till they're already trapping other motorists - but it can be more flexible, reliable and often quicker than a human map reader. It's fun to use and provides information such as estimated times and distances which I've found quite accurate and interesting.

Whatever its aims, QRoute cannot be a perfect source of information, but it can provide useful guidance for a human. I've often found similar programs for other computers entertaining, if not always reliable. The arrival of a route planner for the QL warmed my heart when I found out about it at the Irish QL Workshop, and the name of the converter - former sub-group eminence and CGH Services developer Rich Mellor - boosted my expectations.

Support

QRoute is supplied on one 720K disk, with a clearly-printed A4 manual. The first 27 pages discuss QRoute menus, map editing and configuration, while another ten pages address QPTR extensions, which I found less than intuitive to use. The lengthy discussion of Menu-Config seems gratuitous as there's only one option it can set - the data file location - and a standard CONFIG program is also supplied.

This program was originally written in unstructured Atari BASIC, and some limitations stem from its birth on a non-multi-tasking system. The original version remains available on the Internet, but the Web address given in the manual is wrong. Users in need of supplementary information should pay a visit to:

<http://www.dur.ac.uk/~ded1brh>

- the contents of that site are mainly aimed at users of the original shareware Route Finder for the Atari ST range, but author Brian Henderson of Durham University (email address: B.S.Henderson@dur.ac.uk) says "we hope to update the pages in the near future for QRoute." I was disappointed not to find any more maps, but there is a useful list of major roadworks and bottlenecks which goes some way to address the criticism that QRoute ignores intermittent but serious congestion. Pointing the way?

The required QPTR files were supplied on the disk. I find the Pointer Environment gets in the way of multi-tasking and the sharing of information between real Qdos programs, but in this case it makes sense, because you can point to the displayed map and make selections that way. QPTR should also make it easier to adapt the program to higher resolution displays, although Dilwyn Jones was unable to stretch the current version beyond the old QL limit of 512 by 256 pixels when he tried it on his SuperHiRes QL emulator, though he could drag the display around a larger screen.

Rich Mellor reckons the display should be resizeable on most QL-compatible screens, although he uses a normal QL display, and Roy Wood says he's tested resizing on Aurora and QXL. **[We now know this was a bug in an early version, now fixed - Editor]**

The user interface is a bit crude unless you have a mouse; this could be much improved by positioning the pointer sensibly as keyboard short-cuts are selected. I often found myself pressing ENTER only to find that this was interpreted as ESCAPE because the software had opened a new window for data entry but

left the pointer outside its border, on the Quit button at the top. This also prevented the keyboard short-cuts shown in the adjoining window from being recognised.

For a multi-tasking system the implementation is clumsy; the blame for this must be attached to the original source, single QLiberator task, and the library code that generates the menus. The program is 'modal' so that other buttons stop working as soon as you select one; the menus waste space on the screen and cannot be resized, though you can change the size of the main display.

Part way through its history, a change to QPTR disabled the familiar SuperToolkit 2 ALT ENTER keystroke to recall past entries. I kept fruitlessly trying to recall commands, after running the supplied BOOT program. The undocumented HOT_GO command brought back the facility, although not the previously-typed commands.

This command may be added to your BOOT file, although not until after you've finished loading code into Resident Procedure space, as the new Hotkey system restores ALT ENTER by adding a task which cripples RESPR! One step forward, one step back. But for SMS devotees, and QL users who've taken the step to QPTR long ago, this is no obstacle, and the necessary QPTR files, lacking only a mouse driver, are supplied for the reluctant remainder. Whatever your attitude to mice,

you should not let QPTR stand in the way of the first major new QL application for quite a while.

The only software you absolutely must already have is SuperToolkit 2, either in ROM or as part of SMSQ/E. Unfortunately QRoute runs into problems with the PD utilities PICE and PEX, which attempt to restore the original multi-tasking performance of Qdos to a QPTR'd system, so the manual warns users against using them.

Configuration

QRoute ran straight from the supplied 720K disk, loading the main program in about 20 seconds, but the map takes a very long time to load, even with a Gold Card to speed up processing - more than two minutes. As with other

long-winded operations, a progress bar hops sporadically across the screen as the job is done. Then QRoute laboriously draws the map of Britain, which takes a further 20 seconds, but can be switched off.

QRoute was readily transferred to hard drive by WCOPYing the files to a subdirectory and changing the line 5 dev\$="flp1_" in the BOOT file to 5 dev\$="win1_QROUTE_". This still left it reading the map painfully slowly from FLP1_. You must run two separate configuration programs to cure this. It would be more convenient if these programs were combined.

The standard QJump Config or super-whizzo split-level MenuConfig does the job for references to program files in ROUTE_OBJ, but you can get the same effect by adding a string parameter to the initial EX command. MenuConfig also lets you preset paths for the file request window, a questionable benefit in this case.

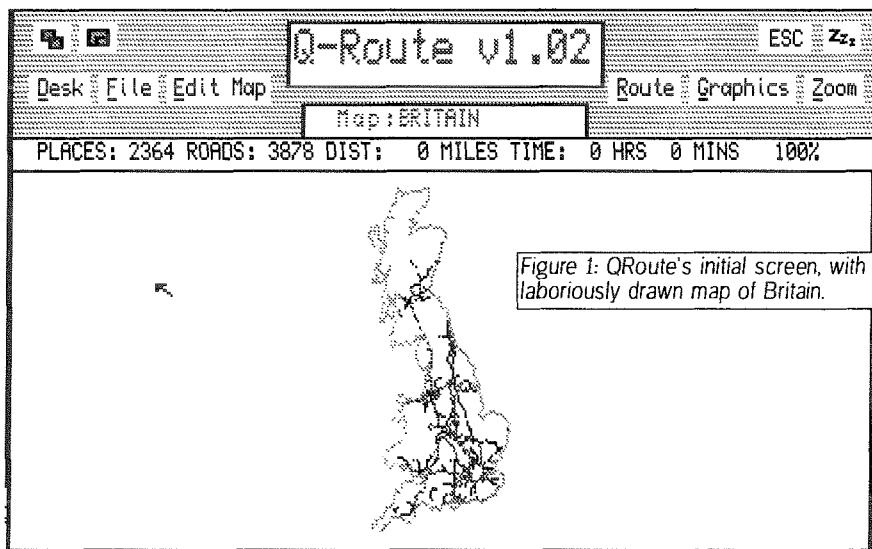


Figure 1: QRoute's initial screen, with a laboriously drawn map of Britain.

A separate program, QRCONFIG_OBJ must be run to tell it where to find maps, where to store output eventually destined for the printer, and many other options. Both the custom configurator and QRoute itself fell over with QLiberator 'out of range' errors when I tried to configure them on hard drive, but worked at the second attempt, when I copied the files on floppy and then to hard drive after configuration. This seems to be a QUBIDE problem.

The specially-written configurator presents a load of options via the pointer environment. Besides default file locations, you can set the display style (graphics or tables) and speed and display detail for each type of road, including colour and type of line, dotted or solid. You can reduce clutter on long-range maps by setting the magnification level at which each feature

appears, and allocate space for additions to the map.

Map loading remains slow on a Gold Card, even after configuration. Loading takes a minute and a half from an IDE hard drive, suggesting

gram might have trouble with areas in conurbations I tried the nearest town, Smethwick. Perhaps to the surprise of Julie Walters, and the relief of Enoch Powell, Smethwick does not exist either, according to QRoute.

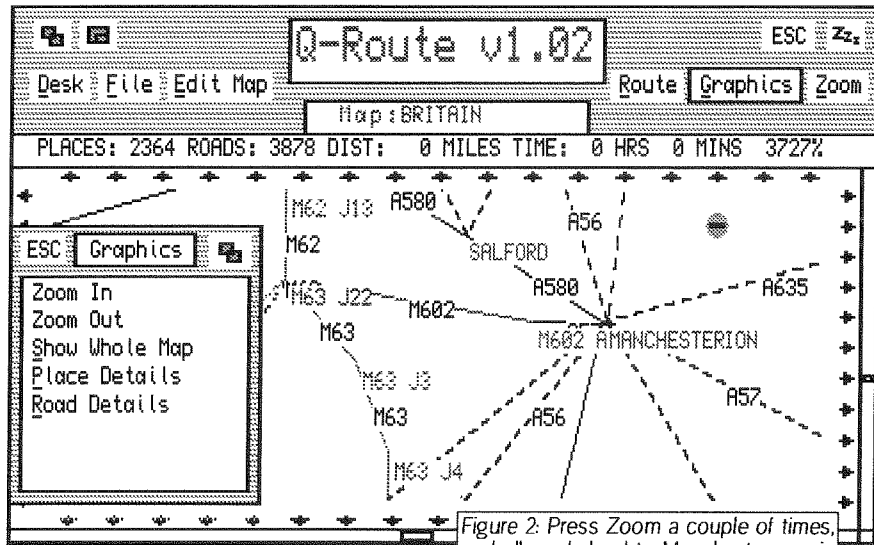


Figure 2: Press Zoom a couple of times, and all roads lead to Manchester - or in this case, 'AMANCHESTERION', due to overlapping text.

that the program code - rather than the device - is the bottleneck, and you'll need at least a SuperGoldCard to bring the delay under a minute. I suspect that a little code or data-format optimisation would make an enormous difference. But once the map is loaded searches are fast because the data is in memory, in the required binary form. The file itself is plain text, much like the contents of BASIC DATA statements, making direct editing possible if you find the built-in editing features too clumsy.

Real world

I live in one of the grey areas of the Black Country - Oldbury, a suburb between Birmingham and Wolverhampton, in the metropolitan borough of Sandwell. I asked QRoute to plot a route from there to my birthplace, Hereford, only to see a message appear: WARNING!! OLDBURY DOES NOT EXIST. Fair enough, I thought, Oldbury may not be on the map, as far as Q-Route is concerned. Thousands of inhabitants might be a bit miffed to find their existence denied, but I'm made of sterner stuff.

I stepped up through my address to the region, Warley. Still no luck - that apparently does not exist either. Suspecting that the pro-

gram might have trouble with areas in conurbations I tried the nearest town, Smethwick. Perhaps to the surprise of Julie Walters, and the relief of Enoch Powell, Smethwick does not exist either, according to QRoute. Each time I inadvertently picked a gap in the database QRoute popped up a list of alternative place names, all starting with the same letter as my original choice. This would help if you made a spelling mistake after the first letter, but I know how to spell my own address. I tried the other side of the road, which falls into Quinton in Birmingham (no house calls please!) but I expect you can guess the response. In fact QRoute refuses to believe in the existence of any places that start with the letter Q, unless you teach it otherwise.

Casting the net wider, I drew another blank with Harborne, which lies just beyond Quinton, but this time the suggestions included Halesowen and Hagley, a few miles out of the city towards Worcestershire. The posh outlying suburbs of Sutton Coldfield and Solihull were among the few areas distinguished from the vast bulk of Birmingham.

I doubt if users in London will be happy either,

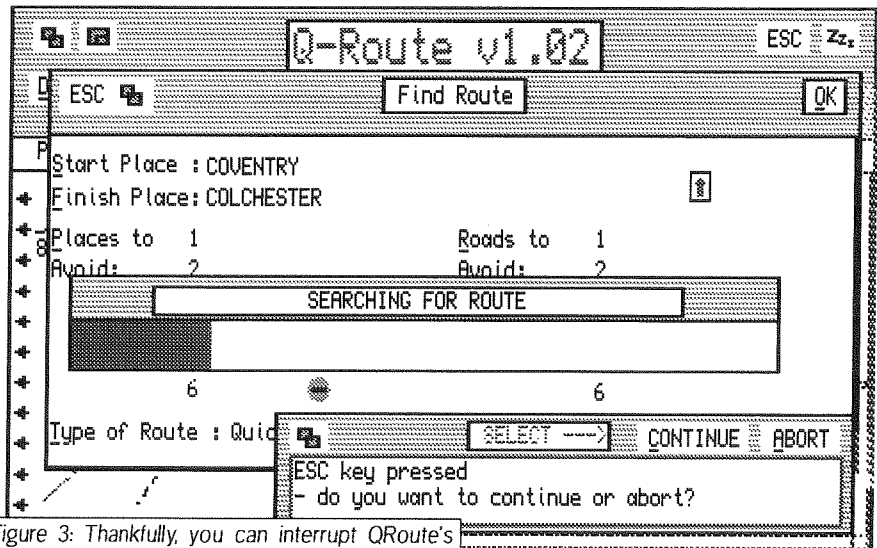


Figure 3: Thankfully, you can interrupt QRoute's searches by pressing ESC.

when it fails to recognise places such as Brixton, Bromley, Hampstead or Westminster. There seem to be big gaps in the program's knowledge where place names are closely clustered, yet these are precisely the places where most potential users must live.

Bizarrely, tiny Woofferton in Shropshire - little more than an ex-USAF transmitter site - gets a mention, presumably because there's nothing else round there and two main roads (the A456 and A49) meet nearby.

QRoute's suggested route from Halesowen to Hereford started out badly, expecting me to start out several miles in the opposite direction,

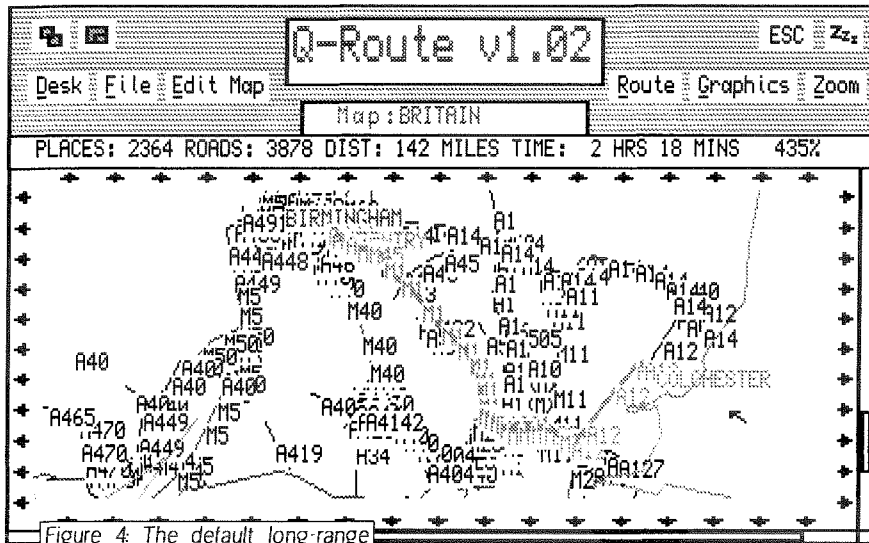


Figure 4: The default long-range map displays get rather cluttered!

up at M5 junction 2 (in the mythical town of Oldbury) when in practice junction 3 is nearer to both start town and destination. So even the places it DOES know are not always spot-on, but once you've found a local landmark, the remainder of the route seemed fine, on my tests.

Considering that the converter, Rich Mellor, comes from Walsall, 15 minutes away (motorway permitting) and he's one of a couple of million West Midlands denizens, I'd have expected more local detail. But it does know about Dudley, Walsall and Wolverhampton, and you can edit the map or make a new one, although a comprehensive UK map would exhaust the memory of a Gold Card, and the patience of many users waiting for it to load.

The advantage of keeping the map in memory is that access is quick during searches, although hardly instantaneous on a Gold Card - it takes a few seconds even to look through RAM for place names, to present a list of alternatives. The new version 1.03 which arrived as I was finishing this review incorporates a machine code searcher which is much faster - to all intents and purposes, instantaneous. Route finding times are also improved, although not as much - sometimes the program only displays half the progress bar before flashing up the complete route - and map loading seems as slow as ever.

I also tried it on QPC 1.21, using a Cyrix 486SLC processor, and was impressed that it ran at all, but frustrated that it took 40 minutes to load the map. Of course this software emulation of SMS on a relatively slow 486 PC is several times slower than any real QL. Route finding took around ten minutes - against perhaps two on a Trump Card QL for a 200 mile trek. QRoute

should be quite usable on a Pentium with QPC, and supports larger displays. I was able to resize the QRoute display to make full use of a 640 by 350 pixel EGA display, but at VGA 640 by 480 resolution the program stuck to QL display limits.

QRoute ought to fly on a 68060, and it booted happily on Amiga Qdos 3.24, currently the only 150 MIPS Qdos system around! Unfortunately the QPTR Toolkit imbedded in the QLib task ran into compatibility problems, reporting an

error from the MDRAW extension, and I was not able to find a way round this.

Test timings

I found QRoute's long-distance suggestions sensible, though rarely exactly what I'd have chosen myself. As a source of ideas for intelligent consideration, QRoute is at its best. I've tried lots of tests; here's one typical example: a route which I have often travelled, from the West Midlands to my Silicon Studio friends in Essex.

QRoute took just over half a minute to come up with a plausible route from Coventry to Colchester - 142 miles, mainly via the M1 and A12. Short routes - twenty miles or so - are found almost instantly, and the 14 hour, 800-odd mile trek the length of the country, from Land's End to John O Groats, is searched out in well under a minute.

The map display (Figure 4) is messy but gives a geographical idea of the route which the Table format lacks. Figure 5 shows the Short Table, which lists road junctions and distances between them. This is your best guide, once on your way. The Full Table is similar but also lists junctions where you drive straight on, without changing roads.

Then I asked for the shortest (rather than quickest) route, and after a similar pause for calculation I was sent down the M6 and the new A14, then south of Cambridge through Halstead

on the A604, trading distance for time. QRoute judged this route about 20 miles shorter but only a few minutes longer.

I tend to stay on the A14 almost all the way to Ipswich, and thence to Colchester on the A12 - but both suggested routes deserve serious consideration. Slight tweaks to the speeds for each road type prompted QRoute to suggest an alternative route heading south from the A14 at Bury Saint Edmonds - another variation I've tried before.

The maps may get confusing unless you zoom in and reduce the number of road and

working with your system you can be sure that QRoute will work too. The new version works with Text87 and Perfection drivers as well. Manual configuration is also possible, for anyone who ENJOYS reading printer manuals!

It's a pity that the printer output file name must end with TXT. The configuration program gets confused if you specify anything else. This is a hangover from the ST, which insists on dots rather than SuperBASIC-friendly underscores in file names. The same goes for the map, and main configuration file FINDER21.INF

The Map

The supplied map of Britain includes about 2,300 places and almost 4000 stretches of road. This sounds a lot, till you try to find ones near your home. It includes 789 villages, 366 towns, 21 cities, 6 terminals and over a thousand junctions of various sorts, including roundabouts, road junctions and full and limited access intersections.

The map is limited to the British mainland: Scotland, England and Wales, plus the nearby islands of the Hebrides, Anglesey and the

'Isle of White' (sic) - a disappointment for Ulster Unionists and Republicans alike - but a separate map of Ireland is in the works, with Germany next on the agenda.

The Channel Islands and Isle of Man do not feature, but routes there are hardly in need of automatic calculation. The latter island was comprehensively modelled - at relatively high resolu-

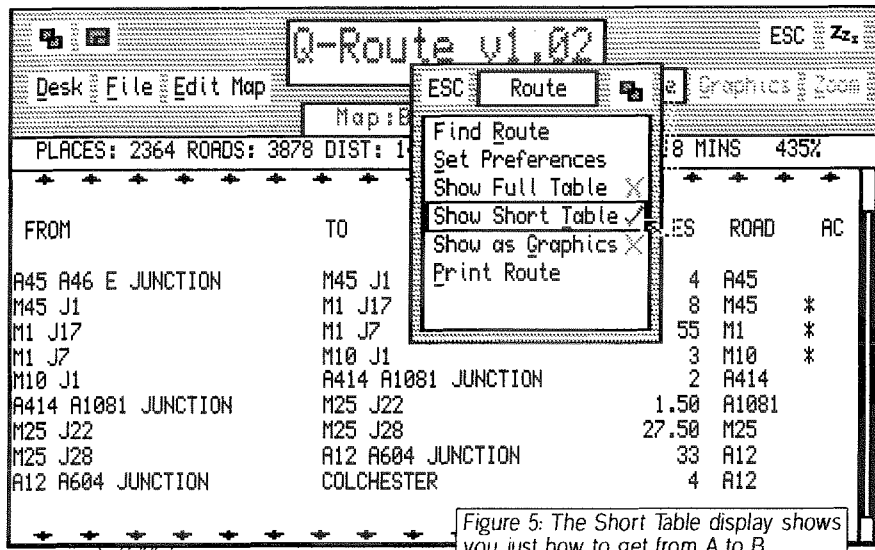


Figure 5: The Short Table display shows you just how to get from A to B.

place names. There are options to control this, but the slow redraw time - typically about 20 seconds per map, on a 16 MHz Gold Card - makes this a tedious process of trial and error, and you rarely get fine enough control to generate a really neat display except for short segments of the route (Figure 6). The close-up display reveals the point-to-point nature of the map. Turns and landmarks - other than junctions - are ignored, giving the map a schematic appearance.

The tabular reports (Figures 5 and 7) are generated quickly and are generally much more useful, especially as they can be printed out, or written to a file for printing or importation into another document. The printer configuration routine cleverly reads a Quill or XChange .DAT file to find the control codes and device for your printer, so as long as you've got the Psion software

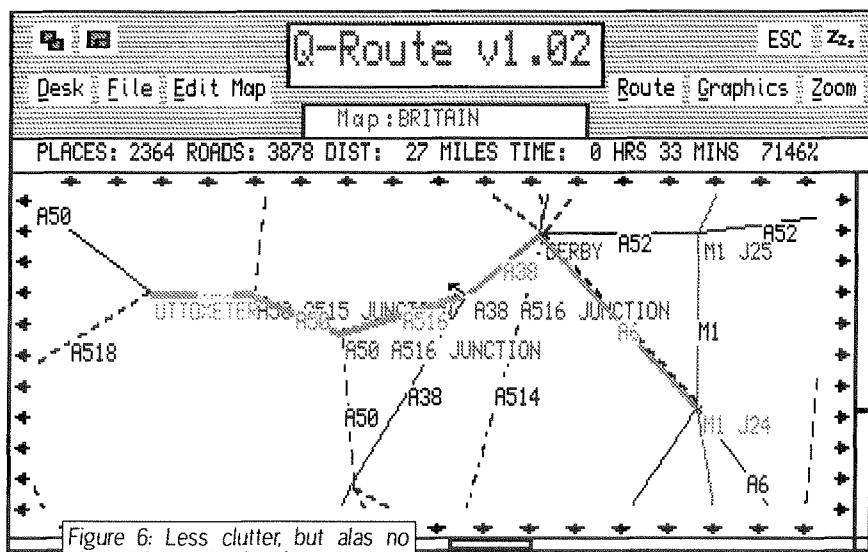


Figure 6: Less clutter, but alas no sign of the Derby Southern Bypass.

tion, too - when it starred as an example file in Talent's TechniQL structured graphics package.

Only major roads and places are included, with no support for one-way roads or local rat-runs. QRoute is a long distance route planner. It won't help you get home before the pizza congeals. The authors realise that the map is a limitation, but it seems generally accurate. Despite mischievous experiments I've not run across serious howlers, except perhaps the total absence of M1 J23A (East Midlands Airport); anyway, users are encouraged to edit the map to add the details they need most.

Map Editing

The manual goes into some detail on map editing, which is just as well. Places are identified by name and a pair of integer co-ordinates, either typed as numbers - rather tricky - or indicated by pointing at the existing map. You can also delete places, or edit the preset names, if you prefer Welsh place names to English, for example.

More essential data link these places by road. The Add Road option requires starting and ending place names, a distance in miles or kilometres, plus the road name and type. Only one route between any pair of places is allowed, and a warning appears if you try to replace a preset route. If a place name is not recognised you can pick it from a menu - thus I resolved a journey from Newcastle to Newcastle by specifying the suffixes 'Under Lyme' and 'Upon Tyne' respectively.

Another route caused more confusion, and took recourse to a real map to resolve it. I planned a route from Uttoxeter, Staffordshire, to Donington, expecting to find the Leicestershire rock concert and motor racing venue, Donington Park. I wanted to see if the program would use the new Derby Southern bypass, recently if belatedly completed.

QRoute started out correctly along the A50, but missed the new road and shot into Derby, then to Nottingham, and off on a long trek into the wilds of Lincolnshire, winding up at Doning-

Q-ROUTE v1.02 by Rich Mellor 1997				
FROM LAND'S END TO JOHN O GROATS				
USING QUICKEST ROUTE				
FROM	TO	MILES	ROAD	AC
LAND'S END	PENZANCE	10	A30	
PENZANCE	BODMIN	45	A30	
BODMIN	LAUNCESTON	21	A30	
LAUNCESTON	A30 A382 JUNCTION	26	A30	
A30 A382 JUNCTION	M5 J31	15	A30	
M5 J31	M6 J8	167.75	M5	
M6 J8	M6 J44	198	M6	
M6 J44	M74 A6071 JUNCTION	6	A74	*
M74 A6071 JUNCTION	A74 A701 JUNCTION	31	M74	*
A74 A701 JUNCTION	M74 J13	19	A74	
M74 J13	M74 J4	29.25	M74	
M74 J4	M73 J3	7	M73	*
M73 J3	M876 J4	8	A80	*
M876 J4	M876 J5	1	M876	*
M876 J5	M9 J9	7	M80	
M9 J9	M9 J11	7	M9	
M9 J11	DUNKELD	43	A9	
DUNKELD	BALLINLUIG	8	A9	
BALLINLUIG	A9 A924 JUNCTION	4	A9	*
A9 A924 JUNCTION	CALVINE	12	A9	*
CALVINE	DALWHINNIE	21	A9	
DALWHINNIE	DAVIOT	51	A9	
DAVIOT	TORE	10	A9	
TORE	WICK	109	A9	
WICK	JOHN O GROATS	17	A9	

DISTANCE= 856 MILES
 TIME= 14 HOURS 0 MINUTES

Figure 7: QRoute's path from Land's End to John O Groats, found in 52 seconds!

RWAP QL Software

All software only available on 3.5" disk
Manuals all supplied in Quill or Text87 Format
All programs need 256K min. unless specified

WARGAMES

War In the East MKII v1.22 (Upgrade from original only)		£10
D-DAY MKII v3.02 (The Allies Take on the Germans)		£15
D-DAY MKII (Upgrade from original)		£10
D-DAY MKII (Upgrade from v1.30-2.13)		£5
D-DAY MKII (PD Version)		£2
Grey Wolf (Graphical Submarine Simulator)		£10

ADVENTURES

Return to Eden v3.06 (Graphics & Text Adventure - 3 Disks)		£15
Nemesis MKII v2.01 (Text Adventure - Upgrade only)	(128K)	£3

GAMES

Open Golf v5.16 (Golf Program - Good Graphics!)	(512K)	£10
Quizmaster II v2.07 (Pub Quiz Program)	(128K)	£5
Quizmaster II (Upgrade from Original)		£2
Quizmaster II Question Module 2 (TV & Films)		£3
Quizmaster II (Original + Question Module 2)		£7

UTILITIES / GENERAL INTEREST

Quick Mandelbrot III v3.01 (Fast Mandelbrot Generator)		£10
3D Terrain v1.20 (Displays Abacus files as Pictures)	(128K)	£10
SToQL v1.29 (converts ST screens to QL format)		£10
Q-Route v1.03 (PD Version Route Finder Program)	(796K)	£2
Q-Route Britain Map (As at 21/10/97 - needs Q-Route!)		£2
Flashback SE v0.01b (Very Fast Database Program)	(128K)	
(Upgrade from Original Only- makes it work on SMSQ/E)		£2

UPGRADES

The version number of the program will normally appear in the boot program, if not, or if the program runs slowly on your system (if it works at all), you probably have the original version of the program. Unless listed above, all upgrades from earlier versions of the programs cost £1 each. Please supply the original disk as proof of ownership. For all upgrades from the original version for D-DAY MKII, War in The East and QuizMaster, please send the original manual.

OTHER ITEMS

I have a range of second hand software and hardware available, as well as some public domain programs. If you want full details of any of the above products or other items, then please send an SAE with a blank disk.

HOW TO ORDER

Contact: Rich Mellor, 26 Ashenhurst Road, Russels Hall, Dudley, West Midlands DY1 2HH
(Payment must be by cheque in Sterling payable to R. Mellor)

If your system can read HD or ED disks, please specify.

ton - between Boston and Spalding. Resort to a real paper atlas revealed at least three Doningtons with three 'n's, and another SIX Doningtons with four, including two just for Shropshire. QRoute knows only one Donington, the one in Lincolnshire, and NONE of the relatively small ones, with four Ns. In fact the venue is known as Castle Donington or Donington Park, but neither appear in the QRoute gazetteer.

In this case I'm using specialist knowledge, after hours scouring maps at Roadwatch to enter obscure sets of temporary traffic lights onto their database, but this does highlight the weakness of QRoute's map and the difficulty of adapting to its limitations. I re-routed to M1 J24, the nearest major landmark, and QRoute offered A50, A516, A38 and A6 - a plausible suggestion if you don't know about the new bypass, but bad news if you were not planning a tour of Derby en route.

So I decided to try to add the bypass. First step was to try to find the junctions at either end - first where the A516 joins the A564 (I think - I'm not typing this at work!) then to the A38 near Findern, and on to the A6, near the M1 junction. These junctions were not currently mapped, so I'd have to break existing connections.

This involves removing an existing route, adding a place between two existing ones, and then adding two routes to restore the connection, probably guessing at the distances. Then do the same at the other end, then join the two. It's possible, but it takes careful thought and is definitely not easy when 'new' roads cross predefined ones.

The slow redrawing of the map at each stage, plus the lack of continuous scrolling and large jumps generated by the scroll bars and arrows at the map edge, made this a very frustrating process. There's no way to move directly to a particular place, short of starting a route there, and no way to merge your own map updates with new ones from QBranch. Rich Mellor has a prototype program to do that, but it's not yet included with QRoute. Apparently maps are drawn in 'only' six seconds on his SuperGold-Card.

Conclusion

Within the limitations of the current map - which is not bad, for long distance planning - QRoute does a commendable job. Its suggestions are helpful and could have practical application, for instance when calculating expense claims. However the map displays are painfully

slow on all but the fastest QL clones, and MODE 4 resolution and colour scheme do it few favours.

Editing new data into the supplied map is problematic, although possible in theory, and the many configuration options sometimes seem an excuse for a rather arbitrary choice of defaults. The map display is clearer on version 1.03, largely because the defaults are better chosen.

QRoute is best seen as an aid to route planning, to be used in conjunction with other information sources - radio, teletext, and roadworks reports in local papers. One of its potential strengths is that you can configure the program to match your experience, adding connections and place names to the supplied map, or replacing it entirely with alternative maps. Such changes are limited, because they don't change the basic nature of the map - a fixed network of nodes, joined by preset routes each of a particular type.

QRoute can't take account of the date or time of day. The time to travel from M6 junction 8 to junction 10A, the most congested stretch of UK motorway, between the M54 and M5 junctions, quadruples as a matter of course in the early morning and late afternoon as commuter Lemmings hit the road, except on bank holidays and mid-weekend. QRoute assumes one speed for each stretch of road, depending on its category, at all times.

The amount of information to do the job comprehensively would swamp any home computer, and require regular bulk updates to make it meaningful. None of the existing systems on other computers offer this kind of adaptive database, though motoring organisations are trying to find a way of feeding live updates to route planners. A keen hacker might be able to tie in live data from the TrafficMaster sensors on motorway bridges, available on the Vauxhall Web page, but you'd have to be an obsessive traveller or fleet planner to make the effort worthwhile.

QRoute is robust and seems well-tested. In use it did not generate any peculiar error messages - though reconfiguration needed a second attempt - and it was compatible with all the extensions in my usual Gold Card boot file. It's easy to use, with a fairly clear display considering the limited QL resolution, colours and fonts. It runs at an acceptable speed on a Gold Card QL, in all but its graphics features - and you don't need those to obtain useful route lists. It's slow and greedy for resources on a standard QL, but it just fits into 896K available

on a Trump Card, and with a smaller map - such as the one of Ireland which Rich Mellor is working on - may even squeeze into 640K.

As it stands, QRoute is an ambitious start, already interesting and useful, but crying out for further development. In some ways it goes further than mainstream commercial packages, with their fixed maps, but it needs extensions - in the form of maps and map-making tools - to develop its full potential. QRoute deserves encouragement, and the best way you can guide improvements is by buying the current version.

At £25 with a full set of QPTR files and a nicely printed manual, the price is reasonable and QBranch and Rich Mellor can be counted on to develop it further if it gains customer support. I hope so, because it's promising and should make great use of the high-resolution high-speed SuperQLs of the future.

Simon N Goodwin, simon@studio.woden.com,
Copyright November 1997

Netted - a quick round up of recent comments from the ql-users mailing list

Roy Wood

If you are on the QL Users maillist that I mentioned in the last issue you will have seen the large and lively burst of communication which was going on at the start of this year. This burst of activity is really what the maillist is all about and what should be going on in the Quanta newsletter, although I think that the main reason it was so lively is that it is instant and people reply as soon as they receive the communication. This makes for a very spirited discussion.

The paperless office took a bit of bashing however because, in order to make the texts available for members of our user group, I printed the whole lot out - all 46 pages of it, including Nasta's epic 9 page reply to Thierry Godefroy. Doing a precis of something like this is always problematic because the emails fly around and get quoted in the replies. Sometimes three or four people will reply to the same piece of text and that is hard to put in any real order on a page. There was a lot more in these messages that I have missed out - either because it got a bit too technical or because I could not fit it into any kind of flow. All the comments I injected into the text when editing it are in parenthesis and

signed 'RW' - well I just could not resist at times. I have tried to keep as close to the text as possible although I have corrected some grammar and spelling from our non-english contributors (the main players, significantly) even though the general standard was very high indeed.

I hope that you find this text both entertaining and interesting - it was a chore to do it but fun at the same time (if you know what I mean). I had to extract it from the emails on my 486 running Windows 3.1 which despite having 4meg of ram and a cut and paste system refused to multitask with a simple word processor and in the end I typed most of this in by hand from the printouts. Some people have asked why we cannot have a cut and paste system on the QL like this, well, you can forget that. On my QL I can 'add to scrap' and don't have to keep running off to the word processor to paste a bit before I can cut another. Maybe Windows 95 is better - I won't hold my breath.

I did not, unfortunately, save the first article in this thread but it started with a request for a way to control the sleep functions on some monitors and the reply by both Nasta and Tony Firshman that there could be a way to do this and it would need some use of the extra lines on the Super Hermes or extra lines being brought out of the Aurora or GF boards.

This quickly spawned request for these lines to be available for other things and amongst them was better sound on future QL developments. Nasta said that the GF was going to use many of the same chips that were to be used on the UGC so anything possible on one would be possible on the other. (The main reason why Miracle abandoned the development of the Ultra Gold Card - RW). This includes MIDI if another small daughterboard was developed and the correct drivers were written (see later).

Nasta also said that many things were possible but wanted to work on the main Gold Fire as a priority. 'Originally a daughterboard expansion was planned which would hold a Motorola 56300 series DSP (and basically nothing else - the 56300 has loads of stuff integrated, including RAM and serial ports to handle MIDI for instance) In effect the card itself could competently handle WAV playback and record (the sound file format used on PC etc - RW).

Claude Mourier replied that a SCSI interface would be better use of these lines and, quick as a flash, Nasta asked what he would plug into it.

Claude suggested 'Scanners, CD Rom drives, Tape Drives Video capture, the usual...' to which response Nasta asked where the drivers for

these miraculous devices would come from and said that he was deluged with people who want these things but have no idea about the complexity of the software needed to implement them.

Andrew Halliwell suggested that the SCSI interface could be a 'very low level interface so anyone with some documentation could be able to perform the operation that is desired. For HD we would need to rewrite the file system so as to get more than the 128Meg limit.'

Nasta said that the problem went deeper than that. 'What we would need is to separate the file-to-sector part of the driver and the sector-to-hardware of the driver. The SCSI driver would only be able to do SCSI type data transfers, which are essentially block oriented and can be directed to and from logical devices. Sometimes the blocks are also numbered, ie: correspond to a physical storage space. A HD driver which would go on top of the SCSI driver would be responsible for converting files to sectors and sector addresses - in effect, that would be the file system, and it would have nothing to do whatsoever with the hardware, at least not directly. Changing the parameters would make this 'file system' driver equally well serve most sector oriented hardware, be it IDE, SCSI, EPP, network or merely RAM or Flash ROM, or even a file or device which is already implemented by this 'file system' driver (anyone remember QXL.DAT files :-)))). Of course, for some devices there would just have to be specialised drivers, but that's a relative minority.

Andrew had also said, '64 bits addressing capability (for a single file!) is already available on modern operating systems (no not Windows 95. It may be modern but it is not an OS!) to which Nasta replied 'Its not modern either. The kernel of QDOS is still a model of modernness as far as W95 is concerned.' He went on to say that he had 'two main objections to the way a QL's file system works 1. You can't make a difference between logical and physical devices by legal QDOS/SMSQ means. This is HIGHLY problematic if you want to implement any of the popular network standards and, for that matter, even the QL's own Network standard. It means you cannot make 'drives' out of subdirectories. If assigning logical to physical was possible any devices you would care to use would just be entries in a directory tree, with storage devices being directories and serial devices being files. That way you would have true device device independence. And, last but not least, the assigning capability would eliminate all sorts of _USE commands which work in a quite dirty way, modifying

system tables that they should not even know about in the first place. 2. There is no separation between file name and file path resulting in the name length limit problem. If only a file name limit was 36 chars that would be fine by me, providing the directory name doesn't count. Making the latter a QDOS-string, 32768 chars, would be perfectly adequate. Since a path could be stored on a 'per job' basis and inherited from the starting job, similar to the Tk2 defaults (which are terribly underused!) then this would eliminate the need to specify the devices mentioned in (1) with the whole path. Besides, given the assigning capability mentioned above, you could always assign 'aliases' of devices deep in the directory tree, to exist in the top level, to make them easily accessible.'

Jerome Grimbert was the next to stick his head over the parapet and make a suggestion about interfaces. Nasta, very patiently, explained why it was very hard to get one driver to access many different pieces of hardware and asked if anyone had read his article on Meta Drivers.

Then there was Thierry Godefroy, whose depth of knowledge was revealed by the full answers he provided to the above questions. He suggested that QXL users who have LINUX (and enough space to put it on their hard disks) could use the drivers provided by that system via a LINUX/QXL interface. He provided a very good basis for his suggestions and jabbed a pointed stick at Nasta by asking why he designed the Gold Fire and not a better QXL (something I have been trying push Stuart into for a while - RW). All of this ended in his volunteering to write a PC/QXL interface.

Nasta, on the other hand, said that 'this will not work anyway if you do not have something to talk to the QDOS/SMSQ end of things and that, in my view, is the main problem. Consider any type of an 'interface device', which by itself really has no function until it is defined by what is on the other end of the interface..... you still need to treat various specific devices in a specific way no matter what 'interfacing' you go through - so in the end you still need a driver... As it stands we can't even do device mounting on as simple things as multiple IDE drives, let alone handle devices whose main purpose is not data storage. If you think WIN_USE is made up in the spirit of SMSQ or the N1_xxxx way of setting up network destinations is made in the spirit of QDOS then you are dead wrong, what it does is very dirty. Imagine (if) you couldn't make any assumptions as to what it is you are accessing at the other end of an interface, like in the case

Professional & Graphical Software

ProWesS

ProWesS is a new user environment for the QL. ProWesS is short for "PROGS Window Manager", but it is much more than that. Apart from a new window manager, it contains all the system extensions from PROGS, and is essential if you want to run programs which need these extensions.

The ProWesS reader is a major part of the package. It is a hypertext document browser. This means that text files which include formatting commands (including pictures) and possibly links to other files can be displayed and read in this program. This is used in ProWesS to read (and possibly print) the manuals, and display the help files. The hypertext documents which are used by the ProWesS reader are in HTML format, the format which is popular on Internet to display World Wide Web pages.

Another important aspect of ProWesS is the possibility to allow programs to automatically install themselves on your system, and to be able to run them without resetting the system. This means that, when you get a new program, all you have to do is insert the disk and indicate "start the program in flp1", a menu option in the "utilities" button. To install a program, you indicate "install software", and the software can be added to your system. This way, you don't need to know how to write a boot file to use the multi-tasking capabilities of your computer.

ProWesS includes many programming libraries. These include syslib, an interface to the operating system, PROforma, a vector graphics system, allowing rendering both on screen and on paper (via a printer driver). The DATAdesign engine is also part of ProWesS. It is a relational database system with a bonus, as you don't even need a key field. You get a powerful record at a time data manipulation extension to the language you already use. Of course it also includes ProWesS itself, the new resolution independent window manager.

PfList

Easy to use program to create listings on any printer (especially inkjet and laser). This ProWesS application allows you to indicate the files which have to be printed. Each column contains a footer which can include the filename and filedate. The listings always allow perforation. PfList can create your listings in two columns and in landscape (or both).

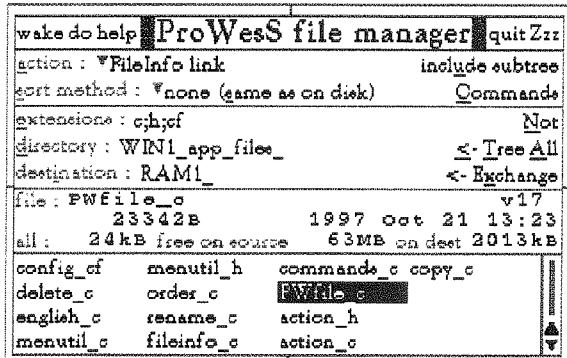
fsearch

File search utility with many useful options, like the choice to search only files with a certain extension, and whether or not the directory tree has to be scanned. All occurrences of the searchstring will be displayed with line number or offset. You can also use special matching features, like case dependent, matching a space with a stretch of whitespace, and searching for a word delimited string.

font- utils

manage your font collection. You can preview fonts on screen, see what characters exist in a font and convert Adobe Type 1 and similar fonts for use in ProWesS.

*New ProWesS application
a powerful and very user
friendly file manager*



LINEdesign

Create artistic drawings, technical drawings, process bitmaps (even scale and rotate them!), and any kind of vector drawings. You can use graphics objects to create the most fabulous drawings ever seen. Because LINEdesign is a vector drawing program, any part of the picture can be moved, scaled, rotated, slanted without any loss of precision or resolution. In LINEdesign, pictures are device independent, meaning that the printout will be the same on any printer (e.g. same size and position).

LINEdesign is good at handling text. You can easily put titles and full paragraphs on the page. All the fonts which are available to ProWesS can be used in LINEdesign.

LINEdesign is a drawing program, but it can also be used by people who are not good at drawing. LINEdesign is a great program for making leaflets, posters, and any kind of printed work. Lots of clipart and extra fonts are available from public domain libraries and BBS's. You can even import Adobe Illustrator files.

DATAdesign

Never before has it been so easy to create, fill in and maintain your personal databases. To start a new file, just type the names of the fields. To add or delete a field, no problem, just do it. To change the name of a field, just indicate it. You can choose which fields are displayed and also which records. You can have a hidden comment for each record, look at the file in tabulated form and transfer data to the scrap or hotkey buffer. Files can be memory based (for speed) or disk based (for safety).

new address !!

Dr. Fr. Hemerijckxlaan 13 /1
2650 Edegem
Belgium

tel : +32 (0)3/ 457 84 88

fax : +32 (0)3/ 458 62 07

email : joachim@club.innet.be

www : <http://www.club.innet.be/~year2827>

ProWesS - BEF 2400

DATAdesign - BEF 1200

PWfile - BEF 900

PfList - BEF 600

Payment terms :

LINEdesign - BEF 1200

fontutils - BEF 1200

fsearch - BEF 600

You have to run ProWesS to make LINEdesign, DATAdesign, fsearch, fontutils and PfList work (even though DATAdesign uses wman).

All our software is normally supplied on high density (HD) disks. However they can be obtained on double density (DD) disks at an extra cost of BEF 100. To use ProWesS and any of our other packages, you need a system with at least 2MB of memory. You should have a harddisk although a two disk system will also work. The use of SMSQ/E is strongly recommended for optimal use of ProWesS.

If you are VAT registered (specify registration number) or live outside the EEC, the amount to be paid is the total (including postage) divided by 1.21 (no need to pay too much).

Payment can be done by EuroCheque in BEF, or by VISA, EuroCard or MasterCard. Credit card orders can be handled by phone. For credit card, please specify name of card owner, card number and expiry date.

Postage : Costs of postage and packaging have to be added. You can choose the quality. Rate depends on no of programs.

copies	priority mail		ordinary mail			
	Belgium	Europe	World	Belgium	Europe	World
one	100	200	240	100	120	145
two	135	340	420	135	190	230
3 or 4	160	560	770	160	310	395
5 to 8	185	870	1250	185	550	705
more	295	1130	1610	295	800	1030

All prices are in BEF, including 21% VAT

of networks or ATAPI or SCSI used by non-storage devices, and, as far as QDOS/SMSQ goes, you have hit a wall no matter what you use to get to the hardware. Any kind of layered device is all but impossible to set up under QDOS/SMSQ unless it is suitably expanded.'

Next came Robin Barker talking about Ama-Sound which someone had mentioned earlier in the discussion on sound cards. 'On my PC I have an 'up-to-the-minute-whizz-bang' sound card fitted only 6 months ago and now pretty much defunct as the novelty has worn off!! The only sound card regularly in use is the 'Ama Sound' which together with a QL provides an alarm warning system (verbal announcements etc) throughout the house (sounds ghastly to me - RW). He too seemed to despair of people ever getting anything done and said 'There are few(er) QL users who are willing to take the time and effort write a letter to their magazine, let alone spend hours/days/months/years developing goodies for the QL... It also amazes me that so many people think that if you develop a piece of hardware you will, as if by magic, have drivers for their digital cameras, tape streamers, computerised Barbie Dolls, external CD-ROMS etc. Reality? who needs it - we have QLs".'

Well that was a strong response from the chairman of Quanta and hopefully - by exposing it to a wider audience - it may stir a few people into writing a few things in reply.

Nasta replied to the above with 'Well I wouldn't go that far but the fact is that most of these add ons are just toys. The fact is also that toying around with them may develop useful tools too. Remember personal computers were considered toys not so long ago'.

Nasta also replied to Thierry's previous communication about LINUX (to write a small SMSQ/E thing that will be responsible for passing the requests to this software interface which will then issue the corresponding LINUX call and pass the result back through the SMSQ/E interface thing.) saying that he wondered why people had not written something along these lines already. "In effect rendering SMSQ into LINUX (?). But then why bother - don't port the programs, run them under LINUX. Or port QL programs to LINUX. I always try to look at it this way. If you have a QXL which cannot work on its own without the operating system on which it sits then it isn't a QL anymore. It is just a question of what you consider to be the essential bit. Why am I saying this? Simply because there are far more things to do than just device access and, even if it was only that, I for

one, believe that the way SMSQ handles things is about 60% approaching **the right way**, which is to say, simple, efficient and elegant. Which is not what I can say about Windows 95! I can say that about LINUX too. In fact LINUX, as most UNIX Operating Systems, is even closer to the ideal in this respect. To me it is obvious that there is no use even trying to make SMSQ more W95 like except maybe in how it appears (graphically thought out) but if we would do things in SMSQ the way they were done in W95 the we wouldn't be here having this discussion - the QL would have died 8 years ago....." "YES, Call me a sick idealist, but if we all weren't why would we continue to do anything for the QL - after all it has been dead for over a decade? Lets face it we are doing it because we feel it has Potential, yes with a capital P. Why I still stick with the original hardware - because with all the other hardware I need to have workarounds" This was Nasta's epic 5 page email which deserves more quoting but space, as always, is limited. Thierry Godefroy's Emails end with 'QDOS/SMSQ Forever' and he added '(and I do mean this)' Nasta replied 'So do I or I wouldn't be doing this'

In reply to Thierry's question about why Nasta did not build a new QXL Nasta's reply was as follows: '1. Miracle was supposed to do that and I'm not Miracle. ... I have the highest regard for Stuart and the fact is that, if it wasn't for his ample help, none of you would have heard of me or would there have been any of my designs around. The harsh fact is that we cannot do everything and there are priorities. Some of us have to make a living out of it too or, if we make a living in a different way the QL ends up on the back burner. 2. I think that the GF will out perform any improved QXL I could ever design. Not by the raw data of how many MIPS (and you know the translation of that - Meaningless Info about Processor Speed) it does but the feel the user will get. It works on it's own turf and does not have to think about interfaces of any sort, nor ways to get around silly limitations of the 'native' OS running on the computer we plug this QXL into. 3. the GF opens possibilities that have been frankly impossible on the QL before without the need to cater for the 99 things to get that desirable 1 out of 100. No, you do not need PCI or 1Gb/sec high speed memory channels as the PC world have you believe. What you need is to use the resources efficiently.'

Thierry wrote 'The fact is that any QL hardware will always be much more expensive than PC hardware so why not use PC hardware?' Nasta refuses this saying that you may only pay

\$30.00 for a graphics card but you also have to pay for the OS that they are designed to run on and the drivers which are part of that OS. This multiplies the figure by considerable factor. (The suggestion has been made both in this list and outside it that if you provide the hardware the drivers for it will appear but, since you cannot test the hardware without the drivers and cannot sell it without testing the writers will have no way of getting it to write the drivers for it. - RW) Nasta, quite rightly, says 'we don't even have 256 colour drivers for the original QXL and the PC which it is plugged into has had 256 capability for ages'

Thierry said that this is because: 1. Tony Tebby did not get paid to write the software (he has to make a living out of programming you know) 2. The QXL to PC interface lacked bandwidth 3. 'I proposed a solution to TT (the 'extended software interface' that would have allowed to make all the drawing on the PC side - no more screen image transfer between the PC and the QXL) but it did not get implemented (yet)'

Thierry also stated 'I prefer to run SMSQ/E on a QXL than LINUX even if it is to say that, in order to get to what we now lack for the QXL, I have to boot up from LINUX first. In fact I don't consider LINUX to be an efficient operating system (for a single user that is) although it is marvellous when compared to all Micro\$oft crap!'

In reply to Nasta's statement that the GF will out perform the QXL Thierry said that was only if the graphic processing is all done on the QXL as it is now - in a scenario such as the he proposed in which the PC processed the graphics we would get the use of the colours and graphic accelerators that were available to the host PC, thus avoiding the bandwidth limitations. The restrictions that were placed on the original software that resulted in the 68040 processing the graphics were because Stuart originally specified that the QXL should be able to run on a 8086!

In one of Nasta's previous messages he used the line 'of course if you cannot live without QL programs...' which Thierry cut out and replied 'NO, I just can't.' He also asked if Thierry wanted SMSQ to become LINUX but Thierry's response was that he just wanted SMSQ to be able to do all of the things that LINUX can do (and more). This is because there are cases where I have to switch from QXL to LINUX (or worse W95, shame on me) to do something that the QXL cannot do (not because of any weakness in SMSQ but just because the program did not get written/ported or because the device driver

does not exist) I really would not mind having a PC with just a limited LINUX kernel (run to the QXL) and then using it only from the QXL (never switching back to LINUX) - in fact this is my ideal.'

He suggests that, if he could get the QXL to run under LINUX it will only take a matter of weeks for the driver to be written. then all C programs using TCP/IP stack (necessary for Internet communication) could be ported to SMSQ/E.'

There was also a brief discussion of the virtues of different flavours of Motorola chip which could be used on a prospective sound board as discussed above.. Dan Barbato asked why not use the MC68EC000 instead of the Z80 suggested by one of the previous correspondents. The reply was that the chip was functionally equivalent to the 68HC000 as used on the Gold Card but a better solution would be to use the 68307 which was cheap, had a full 68EC00 on board and a DRAM controller, serial and parallel I/O lines.

Of course things did not end there. Thierry came back with an equally long email replying to all of Nasta's points with more of his own prodding Nasta to work toward a new QXL (- well, if Miracle won't do it - RW).

At this point in the conversation Nasta suggested a bridge card to plug the GF into a PC (sound familiar? The name QXL Gold swims around in my memory - RW) Thierry is impressed by this concept.

(Money came in to the discussion (well it would wouldn't it ?) with calls from Thierry to consider how much it would cost to upgrade to an Aurora /GoldFire/Superhermes/SVGA system compared with the costs of plugging a QXL into a PC? This is a question that is often raised but my experience is that many people have bought bits and pieces over the years for their QL systems and so a direct upgrade from a bare QL/ Microdrives to a tower case system as above is rare (and mentally daunting as you build your experience with your hardware/software. If you assume that people will have PC knocking around spare just waiting to take their QXLs then that is the cheapest option and buying all these QL add ons is not cost effective but many people like having to get out the soldering iron and I must confess that I still do 90% of my work on a SGC/Aurora system even though I have a QXL in a laptop with twice the memory. I will see if this changes when I get my new laptop and run QPC on it - I'll let you know - This brings me on to another part - RW).

Nasta said 'Believe me, I must have a PC because half my work is done on it' Thierry replied

'That's why some QLers leave the QL world, because anyway they have to do half their work on a PC (this is also why the QXL is a good thing - it helps to keep the users in the QL world and it could help even more if anything - from word processing to Internet connection - could be done on a QXL' Martin Wheatley added 'Most people who are in both the PC and the QL camps are not that way for fun. The QL is for enjoyment and the PC is serious. It is to do work at home or to assist them in getting better at what they do at work. Not everyone is in the fortunate position of Nasta or Thierry and can lay down what the configuration should be. People need to use on their machines what (the programs - RW) they use at work and in the real world that has to be the priority. It has nothing to do with what is best or what is nicest. However Thierry may dislike it, that means the fact that a machine can run Word or Lotus 1-2-3 is important (and yes, that it is the latest version often matters too) LINUX is not really an option for a lot of people. the space taken by QXL is often the most compromise people can make.'

Nasta's answer was 'Exactly - and the fact that it is 'for enjoyment' says a lot about it - it definitely would not be that way if the QL was a chore, awkward and abrasive to the user.'

Brent Hayhoe added, 'I have, almost from day one, used the QL for my business - word processing, invoicing, and accounts. (The same goes for Q Branch and some of our customers - RW). Proudly I can say I still have not got a PC. (This is about to change since my wife needs one for her work) My internet activities done through my clients (splitting hairs her I think - RW) (I am a contract design engineer). Here I am fortunate enough to use a SUN Workstation running UNIX so, when our PC arrives, will be installing LINUX.'

Nasta came back to this with, 'I wish half their work had to be done on a PC! It seems that sooner or later (and more frequently sooner) 98% of their work ends up on the PC. It is all about perceived convenience. Sure SMSQ looks old compared to the flashy colours of W95 but that is beside the point when W95 does not do what you want it to for the first time (mainly by crashing). It is amazing that so many people are willing to live with 'Re-install it all' as the explanation for all of their problems. In my book that means 'we haven't got a clue why it crashed even though we wrote the thing' I prefer SMSQ because there it is at least humanly possible to understand what is going on. I always find it amusing when people complain about QPAC 2

Boot files - just find all the 'ini' files and W95 and let me hear you then. At least the QXL fires up every time - I don't need to worry if it's forgotten where the registry file is or if I shut it down properly last time, or if it will ask me to scandisc all my drives because it crashed the last time all of a sudden - it does not have these 'features'

Nasta said that 'one of the main ideas for the GF is that it should make further expansions cheaper. If I had it before Qubide it would have cost half as much and worked four times faster. Oh and the Aurora would probably be using a PC VGA chip then.... my argument in favour of generic QL hardware, it does not run mainstream PC software, and it should stay that way. Trying to make QL specific hardware abide by PC hardware rules is like making a two year old play the stock market. My experience is that most PC users spend their lives installing and uninstalling and not actually working. I work on a PC and can see that it is not nearly as efficient as I would like it to be. The programs I use are much like the ones in the QL world (except) they don't come up with new versions every 5 minutes. If people would concentrate on working and not installing they would see that they do not need a new version with 2 new options and a sub-sub-sub-menu that they have never heard of but would insist on making the previous version bug free. there is another reason why Qubbesoft/me has not produced a new QXL. TT was paid for the development of the software for the current one by Miracle and we could not just butt in and make a product which would use other peoples software without coming to an arrangement about it - and probably engaging TT to write/rewrite whole sections of it. The Gold Fire is our first product that does not rely on existing or somehow available software. We cannot just use the SGC ROM (even if that were possible). What this means is that it will take more time to be finished - but to get it to the market sooner - we will take it in steps. This is why the on board ROM will actually be Flash Memory so that firmware updates can be done by the user, adding drivers as they become available.'

And it is still going on at the moment as I write this. Davide Santachiara said that 'For the 'very good operating system' I recommend you to read the UNIX-HATERS book at the following URL:

<http://catalog.com/hopkins/unix-haters/login.html>

All of this is just a precis of the messages flying back and forth on this site and I hope that more of you will be signing up to it if you can. It raises a lot of interesting issues and shows what

TF Services

superHermes

A major hardware upgrade for the QL

- All Hermes features (see below for list) PLUS full 19200 throughput on ser1/ser2 not affected by sound
- IBM AT keyboard interface (plus foreign drivers)
- HIGH SPEED RS232 industry standard two-way serial port. 4800cps throughput (supergoldcard - qtpi - zmodem) at 57600bps
- THREE low speed RS232 inputs (1200 to 30bps) Driver for SERIAL MOUSE supplied. Other uses include RTTY/graphics tablet etc
- THREE spare I/O lines (logic) with GND/+5V
- Capslock/scrolllock LED connector
- Turbo/keylock connectors
- 1.5k user data permanently storeable in EEPROM

All this on a professional board about twice the size of the 8049 co-processor it replaces

Cost (including manual/software) **£90** (£92/£87/£90)
 IBM AT UK layout Keyboard **£22** (£24/£23/£27)
 Serial mouse **£11** (£13/£12/£14)
 Capslock/scrolllock LED **£1** (£1.50/£1/£1.50)
 Keyboard or mouse lead..... **£3** (£3.50/£3/£3.50)
 High speed serial (ser3) lead..... **£4** (£4.50/£4/£4.50)

Hermes available for **£25** (£26/£24/£27) Working ser1/2 and independent input, debounced keyboard & keyclick.

superHermes LITE

All Hermes features (see above) + an IBM AT keyboard interface only. Entry level superHermes.
 Cost (incl keyboard lead)...**£53** (£55.50/£51/£53.50)

Minerva

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

The ORIGINAL system operating system upgrade
OTHER FEATURES COMMON TO ALL VERSIONS
 DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if reqd).
 Send disk plus SAE or two IRCs

MKI...£40 (£41/£40/£43) MKIL...£65 (£66/£63/£67)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 including 6 month guarantee

QL RomDisq

Up to 8 mbyte of flash memory for the Sinclair QL
NOW BEING SHIPPED

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent FLASH memory (ie there when the QL is switched off) which can be written to by the QL.

The software to access it is loaded automatically at power up/reset. It uses a directory driver written by Tony Tebby, and logic code from Stuart Honeyball. You can even load ROM images.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off RomDisq at hard disk speed (reading at over 1mbyte per second).

It is an extremely small and compact circuit board, and has hard gold edge connectors, eliminating contact problems and corrosion.

2 mbytes RomDisq.....**£39** (£41/£37/£40)
 4mbytes RomDisq.....**£65** (£66/£63/£67)
 8 mbytes RomDisq.....**£98** (£100/£95/£99)
 Aurora adaptor.....**£3** (£3.50/£3/£4)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors) **£40** (£43/£38/£44)

4 amp total (for motors etc) **£45** (£48/£43/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£23/£27)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required.. **£25** (£28/£23/£27)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temperature measurements, sound sampling (to 5 KHz), x/y plotting..... **£30** (£31.50/£29/£30)

Temp probe (-40°C to +125°C)..... **£10** (£10.50/£10/£11)

Connector for four temp probes..... **£10** (£10.50/£10/£11)

Data sheets..... **£2** (£2.50/£2/£3)

Control software & manual (for all I/F).. **£2** (£2.50/£2/£3)

QL SPARES

Keyboard membrane..... **£12** (£12.50/£12/£13.50)

1377 PAL..... **£3** (£3.50/£3/£4)

Circuit diagrams..... **£3** (£3.50/£3/£4)

68008 cpu or 8049 IPC..... **£8** (£8.50/£7.50/£9)

8301/8302 or JM ROM or serial lead.. **£10** (£11.50/£10/£11)

Power supply (sea mail overseas)..... **£12** (£17/£16/£21)

Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable). Prices are: UK (EC/Europe outside EC/Rest of world). Payment by cheque drawn on bank with UK address, debit card/Mastercard/Access/Eurocard/postal order or CASH! (No Eurocheques). SAE or IRC for full list and details

8 MAR 98

29 Longfield Road, TRING, Herts, HP23 4DG

Tel: 01442-828254

Fax/BBS: 01442-828255

tony@firshman.demon.co.uk

http://www.firshman.demon.co.uk

VISA

MasterCard

side of the fence some people are on. It is good to see people like Thierry and Nasta putting their views and ideas up for inspection and it is altogether possible that something good may come of this - hardware or software or even some understanding of what the people who provide the users with their products have to go through.



QXL matters(?)

H. Huyg

Introduction

There are many roads leading to Rome. Even if we replace the last word with 'Ideal QL-environment'. Personally I think that the QXL path is not a bad one at all: stealing the resources of the PC and still remaining independent. I know very well that there are a great many people who believe that the QXL (or QPC) is just one step toward heresy. Well, I want to know what is going on in the other camp, and living in temptation is more exciting anyway! Just two small examples: the graphics interface on which a number of people have crushed their teeth, and now, what about 'speech recognition', the best thing since sliced bread. I shall not write about the hardware side of the issue.

Acknowledgement

This article is quite critical about some aspects of the implementation of the QXL (of course I mean SMSQ). It might give the impression that I think that some loony guy has scrambled together a bit of everything, with the undesirable result. Far from it: I think that the Operating System, coming from QDOS, and especially the multitasking abilities of SBASIC are great. Knowing a bit about the unspeakable thing, there are some great facilities, and I feel at home with them. Please remember this when I use rude words for some challenges of the QXL.

SMSQ vs. SMSQ/E

There is an Operating System SMSQ, and there is SMSQ/E, which is basically the same thing and running on any platform, including QXL. Presumably the '/E' stands for 'Extras' and that is where the trouble is: I would have to pay the equivalent of DM 199,- to be able to use these extras. I have no idea which part of the QXL's

price is SMSQ, but I have been sold the whole package for a price, not once, but twice (yes, I have 2 QXL's, one for first level backup). Moreover, the /E bit is not fixed, it changes, suppliers have to eat too. So where is the standard base? Being the proud possessor of a QPC (under SMSQ/E), I have taken the decision that I cannot and shall not use any of the Extra features provided through '/E'. This is of course a simple decision, because I could not use them on my QXL! Now, SMSQ/E contains a few commands which to my mind are essential, and I cannot use them! Before I start moaning about SMSQ again, I am going to address the Parameter Passing Problem, which is a general QL problem.

Parameter Passing

Sometimes I have the feeling that the QL community consists of two categories: the 'chickens' and the 'feeders', a characteristic of it all being that the chickens have to swallow what is being fed to them. Unfortunately, I belong to the first group. What has this to do with parameter passing? Nothing at all of course. But, have you ever noticed programs, where after the name of it you can (or have to) put a semicolon (;) followed by some information. A good example is the UNZIP program, you have to supply a file-name, or nothing much happens. Now, have you, a bread and butter user, seen in S(uper)BASIC the command to access all that rubbish? I haven't, but I am a very impatient person, with a very low level of comprehension. What I am trying to say is: there are some 'standard' features in QDOS/SMSQ(/E) just reserved for the feeders. Wouldn't it be nice, if you could write a program in an understandable language, where you could pass a (set of) parameter(s)? Now we come to something, which to my mind is essential, to any human on earth and beyond: "Where do I come from?" If we, for once not concentrate on humans, but on programs, we see that we face a problem. To give a concrete example: G.Wicks got into a terrible scramble (or is it scrabble) with the Style check program. There is this magic program, there are the files the program needs to check YOUR file, of which you were so proud before you ran Style check. Now, where are those files? You can throw everything in the very same directory and then you have no problem, or you have your file somewhere else and then you have. The point is that Style check doesn't know where its rules are. In fact any program needing its own data to be able to execute faces that problem. Is that the reason

why we have a binaries patch program, called CONFIG? Something which should be forbidden? We were not alone: on the PC one faced the same problem for about seven years. However, from DOS 3.x onwards, a program can ask about its origin, and how useful this is: one puts all the files plus programs in its own directory(tree) and 'DATA_USE' is for the user file(s). The QL is about 14 years old. **[Question from the editor: can DOS execute programs from a Thing? What kind of origin would you specify here?]**

System Information

The QXL was (and is) supplied with not one SMSQ, but four, each one coping with one set of screen dimensions (512x256, 640x350, 640x480, 800x600) SMSQ/E allows you to change that dynamically if you are careful, and allows a bigger resolution: 1024x768, but you would need a 43" monitor to be able to read what has been written. The knowledge about these dimensions, and some other 'variable constants' I call 'system information'. The extraordinary fact is that we get supplied with a bit of hardware plus software with all those capabilities, but the chickens have no means to know where their system information is. Thanks to some clever people at QUANTA (Derek Stewart) and the editor of QL-today (Dilwyn Jones), we are finally, in 1997 presented with that information. My point is: should that information not have been supplied immediately, at release time? I remember the articles in various papers, all supplying incompatible 'peek & poke' gadgets, just because the lack of system information to the program.

Unit Management

If you have a QXL/QPC, how do you transfer files between the two systems? Answer: with great difficulty. Now this is amazing. When I bought my first QXL years ago, there was in its documentation a simple one-liner: 'There is currently no way of accessing MSDOS files on the hard disk.' Well, I thought, you cannot have everything all at once, let's wait for it. I have given up waiting. I have written a program which can transfer files between 'WINx_' and 'c:\' and the other way around. However, on the QXL we are facing a very annoying problem: on the QXL namely, you can switch between PC-running and QXL-running in a very quick and often a dirty way. You would say: "That is just what I need: I

dictate all this nonsense to the PC, pass it to the QXL, make it a Quill file and send it off to QL-today. Unfortunately, I am not a perfect person, I need not one, but several cycles to have a printable (?) article. And now we face the identical problem as with the floppy: - write file A on it from PC, - start SMSQ, - read file A on QXL, - switch to PC, - write file B on it from PC, - resume SMSQ, - where is file B? In other words: I have to start from scratch SMSQ when I come from the PC. Resume takes a fraction of a second, starting from scratch takes ages, and my working environment is set to the beginning, i.e., I have to set that up as well again. With the hard disk this is much more dangerous: it is very simple to ruin a complete 'WINx_' unit. I have found a way to limit the damage. Let us come back to the quick and dirty way of transferring between the two systems: - from PC to QXL is under program control, - from QXL to PC is under keyboard control. The latter way is unbelievable! Why is that not under program control? At one place where I worked there was always a 'leaning on the keyboard' test when a new application was to be installed. This is as bad as CTRL-ALT-DEL on the PC! Why is this random switching dangerous? For obvious reasons, it is very expensive to check with each I/O-operation whether a common resource has changed (by an alien). However, the 'WINx_' unit contains a facility in that there is in the 'Volume Header' an 'Update Count', which, as far as I can observe, is increased each time there has been a change in the 'layout' of the unit (new files, new sectors, ...). So in theory a 'Resume' could check this information on the hard disk, and if different, generate an error whenever that unit is accessed subsequently. As you can imagine, all that is quite tricky and it would be much better to allow switching only at controlled moments (still having above check and action!). Somewhere in some documentation there is the suggestion of the 'DEL_DEFB' command, but that closes ALL units on the system, even the 'RAMx_' units, that is chasing the Devil with Beelzebub! For my infamous Utility Program (Q-Cruiser) I would be extremely happy if there is a way to 'Close' a specific unit, so that a subsequent access will read the required control information from the physical medium. This would solve the floppy disk transfer problem as well!

Switching QXL & PC

Everybody with a QXL knows that if you switch from QXL to PC by pressing those

horrible keys, that the QXL continues happily, until it meets a common resource (keyboard, screen, disk), then it stops, and when you say 'Resume', it carries on from there. This is great for calculating PI with many decimals, or finding the next prime number. My system runs often with a program in the background, coming alive every 15 minutes or so, accessing the disk. If I have been so unlucky to have switched to the PC in the meantime, then the QXL crashes. I have a suspicion that disk access or something else is not completely checked when switching.

Pointer Environment

Now we come to the 'piece de resistance' and I am going to make many enemies. I do not understand it at all. There is this great effort to be as far as possible away from anything what even looks like a PC and there is at the same time this massive push to simulate Windows! And all this with graphics of the year one! Maybe I see it wrong, but I associate Pointer Environment with that dreaded animal: the mouse. What do we see on the PC? That thing is so littered with meaningless 'icons' that nowadays it tells you in plain words what those silly little things mean! But the basic message is: 'How can you work with one hand on the mouse and the other on the keyboard?' Anyway, I notice that I am not the only person having difficulties with such a setup. But there is light on the horizon! This document has been prepared on the PC, not using the mouse, nor keyboard, but as I SAID before, by talking to the electronic thing. You might wonder what the QL's role then will be. The answer: pursue what YOU want to do with it. The current trend in the PC world, and unfortunately in the QL-world too, is to make programming so complicated that it is only for the feeders and I want to chicken out, and yes, I know how to. But that will be the subject of another article.

■

RWAP QL Software - who are they???

Rich Mellor

As some of you will have noticed from the Jan/Feb issue, I have now decided to set up in business on my own, selling some of the older

software titles in which I played a part. There are several reasons for this, including the fact that a large number of my programs have not been published for some time and some have never officially been released for various reasons.

Well, following the successful launch of Q-Route (sold by Q Branch) filled my coffers with a little money, I thought that now would be an opportunity to invest some of the royalties raised in re-releasing some of this older software, which fills a niche left in the market by the other software suppliers.

First of all, I have a plea or two - I would welcome news on the whereabouts of Mark L. Stueber (former proprietor of Sharps Inc. who originally created War In the East), and also of the authors of the Nemesis text adventure (Paul Brittain) - this was sold in the UK by Talent.

Finally, I thought that you would all welcome a little news of the progress of the various programs which I sell:

D-DAY MKII - this is now at v3.02 - as well as being much quicker than the original program, v3.00 incorporated much enhanced computer intelligence, so if you have an early version of this wargame, I would highly recommend updating. The latest version also incorporates a means of allowing the program to run from sub-directories. A PD demonstration version of this program is also available.

War In the East MKII - This is a much enhanced version over the original and has never before been released. It incorporates a lot more computer intelligence and all of the bugs in the original have been fixed, as well as the general program speeded up. There is even now a two player mode included, as well as the option to play all three scenarios as if they followed on from each other (ie. you start the next scenario with your armies in the state they were in at the end of the previous scenario). Unfortunately, as I have been unable to contact the original author, this can only be offered as an upgrade to the original version.

QuizMaster MKII has been amended to allow it to work on much faster operating systems and in MODE 4. There are currently two sets of 300+ questions and I am working on other sets.

Nemesis MKII is again only available as an upgrade from the original version at present (until I can trace the original author). This has various bug fixes which stopped you completing the original adventure (or even getting the first part completed).

■

Glossary of Abbreviations and Terms: S ... Z

Dilwyn Jones

INT - Interrupt or Integer. An interrupt is a signal to a microprocessor within a computer

SIMM - Single Inline Memory Module, a type of memory card used by PCs and possibly by the Goldfire expansion unit when eventually available.

SMSQ - New and enhanced operating system for the QXL, from Miracle Systems. Unlike SMSQ/E, this does not include the Pointer Environment. The letters SMS were never well defined, some say it stands for Single-user Multitasking System, while others say it stands for Small Microcomputer System, and others say Smart Micro System!

SMSQ/E - Extended version of the SMSQ operating system for the QL. This version comes with the equivalent of the pointer environment files PTR_GEN, WMAN and HOT_REXT built in and offers a large number of additional features over SMSQ, e.g. additional devices and device features, device buffering and the ability to change display resolution.

SS - Single Sided. Refers to a type of floppy disk, or its drive.

SuperBASIC - The QL version of BASIC. This was designed by a lady called Jan Jones for Sinclair.

SVGA - Super Video Graphics Array. A PC graphics card. On the QXL or QPC an SVGA mode implies a display of size 800 pixels across and 600 down.

SW - Shareware, a method of software distribution where the author lets you use either a cut-down version of a program, or a time-limited piece of software. If you like and wish to continue to use the full version of the software, you are expected to contact the author and pay a fee, for which you'll sometimes get a non-limited version of the software and product support. Sometimes used as an abbreviation for "software" only.

TCP/IP - TCP (short for Transmission Control Protocol) IP (short for Internet Protocol). These two are usually used together as an TCP/IP 'stack'. It is called that way because TCP builds on IP (hence the actual abbreviation is 'TCP over IP'). It is called a 'stack' because IP is built to work on the actual hardware which

implements internet communications, TCP is built on IP to provide advanced communications features, and possible applications are built on TCP to allow the user to use the communications capabilities, in a layer-like fashion. TCP is protocol between hosts in packet-switched computer communication networks, and in interconnected systems of such networks. The TCP part (or 'layer') handles Data Transfer, Error detection and correction, Flow Control, Multiplexing (emulating several communications channels over a single medium), Connecting to hosts, and Security aspects of communications. IP on the other hand is a protocol which handles how packets are distributed over packet switched networks, including addressing (where the data is sent and from where it's received), transmitting and receiving, and packing of non-packed, or stream data. Usually the IP layer handles the actual hardware which is used for internet communication, although there might be another layer in case different hardware can be used (eg. telephone lines, Ethernet, etc.) *[thanks, Nasta]*

THING - Horrible term for a general purpose facility built into the enhanced QL systems with pointer environment etc. The designers found it hard to give this facility an accurate name due to the general nature of the beast, so they called it THING. That same generality makes it hard to describe in simple terms what a Thing is. The closest we can get is that it is a part of memory with a name of some kind. This part of memory contains a facility of some description (it may be a menu, an extension, a routine, a program, data and so on). The operating system maintains a list of these 'things' and a programmer can use them by looking through the list for a thing's name, and call it as required. Don't worry too much about things as a user - the QPAC2 manual says that 'things rarely go bump in the night', you can manage quite well without having to fully understand them. Anyone who used Michael Crowe's QL MegaToolkit will know that it provided a broadly similar facility called a WOTSIT. Who chooses these names, I wonder?

TK2 - Abbreviation for Toolkit 2, a commonly used set of extensions to SuperBASIC providing additional 'words' to enhance the BASIC language understood by the QL. It was originally written by QL guru Tony Tebby and available as a plug in EPROM chip for the QL. Nowadays, it is commonly built into expansion cards such as the Trump Card, Gold Card and Super Gold Card, and also included with the

SMSQ versions of the QL operating system. If your system does not have a copy of this (only older systems, or unexpanded systems are likely to suffer this) it is well worth getting one.

TLA - Three Letter Acronym, such as BTW for Bye The Way.

TT - (i) Tony Tebby, QL designer and guru.
(ii) A version of the Atari ST, for which you can get a QL emulator.

ULA - Uncommitted Logic Array, a type of logic chip.

UNIX - UNiplexed Information and Computing Service, an operating system written by Ken Thompson of Bell Labs.

uQLx - A shareware QL emulator for Unix based systems. See QL Today volume 1 issue 4 for more details. The author, Richard Zidlicky, can be contacted by email at rdzidlic@cip.informatik.uni-erlangen.de

VGA - Video Graphics Adaptor for the PC. On the QXL, for example, a VGA display refers to a screen mode 640 pixels wide by 480 pixels deep.

WMAN - The Window Manager. This is part of the Pointer Environment (or Extended Environment). Provides a set of menu and display routines which a programmer can access to ensure that programs have a 'standard' appearance, or programs which look consistent with each other. Always used in conjunction with PTR_GEN (see above).

WORD - Unit of computer memory. A Word is 2 bytes, or 16 bits of memory. Can store numbers from 0 up to binary 1111 1111 1111 1111.

WORM - Write Once, Read Many times. A device which can only be saved to once, and from then on only read, like a CD.

WP - Word Processor. Also the term used for the USA equivalent of Quill.

WWW - World Wide Web - a layer of the Internet, devised in Switzerland some time after the basic Internet came into domestic use.

WYSIWYG - What You See Is What You Get, normally used when specifying how close what you see on the screen will be to how it would appear when printed on paper.

ZIP - (i) Term used for a commonly used program to compress files into a single large archive.

(ii) A type of removable cartridge mini-hard disk or superfloppy, made by a company called lomega.

I hope you find this list useful. If you come across any other commonly used QL abbreviation, let me know and I'll add it to the list and update it from time to time.

TTFN (Ta Ta For Now) - Dilwyn Jones



The use and abuse of floppy disks and drives - Part 2

Mark Knight

3. Maintaining your disk drives and disks

From comments in the last issue you will gather that the polish on the drive heads and smooth finish on disks is important in allowing the read/write heads of the drive access to the magnetic domains of the disk. This in turn means that keeping your disks and drive heads clean is very important. It helps not to smoke in a room where disks or drives are stored or used. Using very cheap disks is bad news too as they wear more, thus leaving some of their magnetic coating on the drive heads instead of keeping it on the disk where it belongs.

Cleaning disk drive heads is a subject that causes people to talk an awful lot of rubbish, especially some of the makers of cleaning kits. If you use a floppy drive every day on your QL and don't have a hard disk drive the floppy drive should not be cleaned more often than once a fortnight, and it is better to do it once every four to eight weeks. If you smoke or permit smoking in the room where the drive is kept then double this frequency or you may allow the heads to become badly coated. If you only use really good quality branded disks then clean the drives less often as the better brands of disk shed far fewer magnetic particles: once every eight to twelve weeks is plenty in this case.

The usual method of cleaning is to put in a cleaning disk sprinkled with a little isopropyl alcohol and try to format it several times. Ignoring the "format failed" or "read only" message take out the disk and give the drive a couple of minutes to allow the alcohol to evaporate. You may wonder why I suggest not doing this too often: it is because these cleaning disks with their cloth or fibre surface will wear the

SINCLAIR

Z88

OFFICE/FAX 01494-871319 (EEC) *W.N. Richardson & Co.*

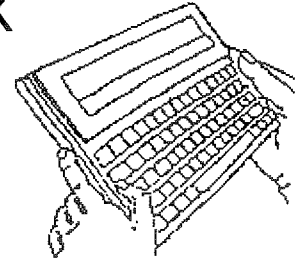
MOBILE: 0850 597650

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL

THE CAMBRIDGE Z88 A4 NOTEBOOK

WITH BUILT-IN WORD PROCESSOR, SPREADSHEET, DATABASE, BASIC, CALCULATOR, CLOCK, ALARM, CALENDAR & VT52 TERMINAL.
USES 4XAA ALKALINE CELLS (ca. 20 HOURS) *



WE DO REPAIRS, PART EXCHANGES, AND WILL BUY Z88'S & PARTS

DESCRIPTION	UK	USA
CAMBRIDGE Z88 ISSUE4-128K	£120	\$180
CAMBRIDGE Z88 COMPUTER	£99	\$130
RECONDITIONED Z88	£60	\$90
32k RAMPACK	£18	\$25
128k RAMPACK	£38	\$50
32k EPROM PACK	£18	\$25
128k EPROM PACK	£24	\$33
256k EPROM PACK	£55	\$75
EPROM ERASER	£32	\$43
PARALLEL PRINTER LEAD	£39	\$50
SERIAL PRINTER LEAD	£12	\$16
* MAINS ADAPTOR (230vac; 6v,500ma)	£10	
TOPPER (PROTECTIVE COVER)	£12	\$16
CARRYING CASE (PLASTIC)	£10	\$12
FILE TRANSFER TO OTHER COMPUTERS		
SPECIAL Z88-QL SERIAL LEAD	£10	\$15
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	£2	\$3
PCLINK KIT (For PCs) }	£25	\$35
Z88 TO MAC KIT }	£25	\$35
Z88 TO BBC KIT }	£25	\$35

**1/2 PRICE
STARTER PACK
1 - Z88 PLUS
QL TRANSFER KIT
+ 32K RAM
+ 32K EPROM
£72 ONLY!!**

POSTAGE UK UP TO £5. EEC £15. USA £20. OTHER COUNTRIES £30.

ALL THE STOCK IS WARRANTIED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING ARGREED, BUT THE ITEM BEING OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

QL & PC COMPUTER USERS WILL FIND THE CAMBRIDGE Z88 ESPECIALLY USEFUL FOR WORK AWAY FROM THE DESKTOP, WITH TRANSFER PROGS DATA CAN BE SAFELY EXCHANGED WITH THEIR DESKTOP SYSTEM.

W.N.RICHARDSON & CO CONTINUES TO PROVIDE FULL SPARES AND SERVICES FOR SINCLAIR COMPUTERS, QL & THE CAMBRIDGE Z88

drive heads much more than any floppy disk (even one of terrible quality). So cleaning it too often will shorten the life of the drive almost as much as not cleaning at all.

Cheap disks shed more of their magnetic particles, as stated above, though they tend to do this mainly when new. This might seem odd but the resin that the magnetic particles are embedded in is quite good at holding on to particles inside itself even in a cheap disk. One area where cheap disks often fall down is that the surface polish isn't very good when they leave the factory and the particles are not properly held in the surface of the binding resin. This leaves it up to your disk drive to put the final polish on the surface of the disk and rub off the loose magnetic particles...

Another area where good branded disks often score over the cheap ones is that they keep data for longer and last much longer. Quality disks generally have slightly smaller magnetic domains and better chemical stability in both the matrix and magnetic particles. This means that while cheap disks may perform well when new or when in everyday use they don't perform well if kept in a cupboard for a while. A box of really good branded disks with files on can be kept in a cupboard for years and will still be 100% readable: this is much less likely with cheap disks.

As with all man-made materials the coating on a floppy disk will eventually deteriorate due to wear and chemical attack by the atmosphere. Excessive heat, moisture or cold can all play a part in accelerating this deterioration as can dirty or dusty environments including cigarette smoke. Keep disks clean, dry and at comfortable room temperatures as much as possible and you should get a good long life out of them.

4. Formatted and unformatted capacities

Often you will see disks rated by both their formatted and unformatted capacity. DD disks for example are quoted as having an unformatted capacity of 1Mb yet a formatted capacity of 720k which is less than three quarters of the unformatted capacity. In fact the unformatted capacity is always 1Mb while the formatted capacity may vary depending upon which operating system was used to format the disk. MS-DOS and QDOS both have a formatted capacity of 720k on a DD disk while the Acorn DFS (disk filing system) used on the BBC model

B formats DD disks with a capacity of 800k. ED disks formatted under MS-DOS have 2.88Mb capacity yet under QDOS they yield 3.2Mb.

What is the significance of unformatted and formatted capacities? To answer this question we must return to the manner in which data is recorded on a floppy disk. All data is recorded as "bits", that is one of two alignments of the magnetic domains of the disk coating. One alignment is used to represent "one" bits while the other represents "zero" bits. Switching from one alignment to the other can only be done at a limited frequency, hence a minimum wavelength. The unformatted capacity is the theoretical limit of the disk if all the bits on each track are used, end to end all the way round the track; however we can never record data on the disk in this manner.

If we try to record data as a continuous stream of bits we present ourselves with a problem trying to read it. Computer data is organised into bytes of eight bits, but if there is a continuous stream of bits all the way around a track how would the drive software know which bit to use to start the sequence? To use a fixed coded sequence would mean reading anything up to a whole track just to find the start, and then we would need to make sure that sequence of bits never occurred in any data written to the disk.

There would be another problem because if we use exactly the theoretical limit of the disk at the standard rotation rate and ideal recording wavelength we will have to spin it at exactly the right speed every time we put it into the drive. If we don't do this the last bit might overwrite some of the first bits on that track or there might be a gap. Also other drives would have to match the exact wavelength when reading the disk and this is impossible even with the very best modern engineering: we can get two different drives to work at very nearly the same speed, but even a single drive will vary its speed a tiny bit from one occasion to another.

To overcome these problems each track on the disk is recorded as a series of "sectors", little streams of data with gaps between them known as "inter-sector gaps". The inter-sector gap is not fixed by the drive but by the operating system used to format the disk. This means the inter-sector gap can be varied by software controlling the drive, and so can the length and the number of sectors.

The gaps give the system some leeway to allow for different drives spinning at different

rates, temperature effects speeding up or slowing down the drive etc. Before the start of each sector is a small "sector header", a stream of data giving some information about that sector, like its sector number, whether it is empty or part of a file, the number of the file etc. All the data in the sector header is decided by the operating system not by the drive.

This control by the operating system allows MS-DOS and QDOS to put 9 sectors per track on a DD disk while the Acorn DFS mentioned previously records 10 sectors per track by using smaller inter-sector gaps. On HD disks MS-DOS and QDOS both put 18 sectors per track, while on ED disks MS-DOS puts 36 sectors per track and QDOS puts 10, but uses 2,048 bytes per sector instead of the usual 512. Having only ten larger sectors means fewer inter-sector gaps which is where QDOS finds the extra disk space. The device driver fools most software into thinking there are four times the number of sectors of the usual 512 byte size.

5. Sectors, clusters, blocks and maps

Sectors on a disk usually occupy 512 bytes whatever operating system is being used and QDOS is no exception. The sector header may contain data about which file that sector belongs to, which part of the file it is and other data such as whether the sector is empty, damaged etc. However this is not enough to make good use of the floppy disk, since in order to load a file we don't want the system to have to search the whole disk to find all its sectors. To find a small file taking up two sectors we might have to read most of the 1,440 sectors of a DD disk and this could take quite a while.

In order to prevent this operating systems label each sector with a number of its own, and also make a table in a fixed position on the disk to show how the sectors are used. On the first sector of track 0 the system will store the first part of the MAP, an area of the disk that contains data about what each sector is used for: this uses sector and file numbers not filenames.

If the disk was simply divided into sectors we would need to store 1,440 entries in the map, which would make it rather large. So just to keep a record of what was on the disk we would fill quite a few of the sectors on the disk with map data. To prevent this the disk is used up in "clusters" or blocks, each of three sectors.

This means on a 1,440 sector disk we need to store data for 480 clusters which obviously takes up less space. As long as the sectors are used up in regular groups of three everything is fine.

Since we have a map file on the disk and we are using the disk up in blocks of three sectors the presence of the map means a perfect disk still has sectors used after formatting as some are reserved for the map. In addition there is a hidden file on the disk where information is stored about each normal file, such as its filename, length, filetype and other information: this is the directory file. When you format a DD disk and get 1440/1440 sectors this tells you that all the sectors are good, but when you ask for a directory straight away you will get a reduced available sector count. You get 1434/1440 sectors; there are two clusters of three sectors each already used, one for the map itself and another for the still empty directory file.

On DD and HD disks there are three sectors to a block (or a cluster, whichever you want to call it), and sectors are 512 bytes long, so the disk is used up in blocks of 1,536 bytes ($3 \times 512 = 1,536$). A file of zero length still occupies 1,536 bytes of the disk and as files expand they use up disk space in further blocks of 1,536 bytes. Any unused bytes at the end of a cluster are simply filled with zero bytes.

On an HD disk there are far more sectors and so far more clusters, so the available sector counts on blank disks are reduced still further; the map has to be larger. On ED disks the clusters are one big sector (2,048 bytes) each so lots of small files on an ED disk may take up more space than they would on an HD disk: with the extra capacity of an ED disk this may not matter. If you have two HD disks filled with large files you may easily copy them to one ED disk and have room to spare but if there are lots of small files filling the same two HD disks they may not fit on an ED disk; this is because of the extra unused space at the end of each file. Files use ED disk space in blocks of 2,048 bytes instead of the usual 1,536.

In fact the ED device drivers are playing tricks on the system to make it think sectors are the usual 512 bytes. When you read the directory on the screen it says a disk formats to 6400/6400 sectors and the first directory reads 6384/6400 sectors. The missing 16 sectors are really 4 sectors of 2,048 bytes each, three for the map and one for the empty directory. The trickery is to allow the disks to work through

the standard system routines which were really designed for microdrives with 512 byte sectors and no clusters (clusters of one sector is another way of looking at it).

On MS-DOS systems the equivalent of the QDOS map is known as the FAT (File Allocation Table) and is organised differently: this is one reason why QL users need special software to read and write PC disks, as they do Atari disks and others from "foreign" operating systems. With QL DD and HD disks the sector and cluster layout was deliberately made the same as MS-DOS to make it easier to write programs to read MS-DOS disks. We can't run programs written for foreign machines on the QL directly but it is often good to be able to read public domain text files or other data from their disks.

In the next issue, we will have a look at harddisk.

■

Beginners Club

by *Andrea and Paolo Carpi*

Greetings from Italy!

The Beginners' Club of Vercelli (Italy) was born as a free association in 1990 from a group of few impassioned after the breakup of a preceding computer club.

After an initial period occupied with the organization of the club and with the restructuring of what is today the social center, it has begun its activity, developing some shareware and free-ware software products for the Sinclair QL in distribution through the Fidonet (and some also on Internet).

Shortly afterwards, it began the work of constitution of a new sector devoted to motor sports. In this field the information technology sector has collaborated with the development of software (always for QL) for managed classifications of sporting competitions.

In the meantime the same information technology sector participated at different national QL Meetings (Italian) and international (Bielefeld, Eindhoven) distributing their own software.

Subsequently (beginning in 1996) it organized a gathering in Vercelli of Italian QL programmers with the patronage of the QItaly Club. Lastly was born the telematic section, thank to belonging to the information technology section

and to new partners for the constitution of the actual BC BBS (2:334/210), a BBS connected to the Fidonet circuit.

The president of the club is Franco Aiazza, author of some programs distributed by the BC. His collaborators are: Paolo Biasutti, vice president and sysop of BC BBS, the brothers Paolo and Andrea Carpi, authors of various software, and Luigi Genovese, who for reason of distance (he lives in another city) is able to collaborate in only a small way.

The club was born to gather local enthusiasts whose hobby is the information technology and since it has developed interesting software, it decided to make it available to all the QL community. We are, however, happy to have aroused an interest in our comparisons in international circles and would be happy to have collaborators and correspondents from foreign countries.

The registration to the Beginners' Club costs 50,000 Italian Liras, but since it doesn't give any particular advantage to possible collaborators from foreign countries (sees the local character of the club), we don't maintain that it is necessary to carry out this formality for whoever had taken an interest in our operation.

In the course of our existence the Beginners' Club has produced the following software (all freeware except The Reader 1.23):

Il Papiro Perduto [The Lost Papyrus]: (PAPIRO.ZIP) Authors: Andrea Iannotti and Franco Aiazza. Small adventure text only in Italian. It is a port and an adaptation of a PD program written originally in GWBASIC. The text of the adventure is entirely by Andrea Iannotti; the adaptation is by Franco Aiazza.

Archigest 3.0 (ARCHI30.ZIP) Author: Luigi Genovese. Simple database in compiled Super-Basic in Italian. For whoever finds it difficult to use Archive.

QL to XMage 0.03 (Q2X003.ZIP) Author: Andrea Carpi. Utility for converting from black and white Mode 4 QL screen to Xmage format and vice-versa. The XMage is a graphic format invented by Enrico Maria Giardano (a Spectrum user from Rome) to exchange images between the Spectrum and other systems. Currently there also exists converters for MS-DOS.

Screen Converter 1.31 (SCCON131.ZIP) Authors: Paolo and Andrea Carpi. Old and slow utility for convert standard QL screens from Mode 8 to Mode 4.

Beginners' Pointer Utilities 2.00 (BPU200.ZIP) Authors: Andrea Carpi and Franco Aiazza. Four

small utilities for Pointer Environment. A menu "Launchpad" style, an utility for overclocking the Gold Card to 24Mhz, an utility for saving the secondary dictionaries of QTYP automatically and an utility to activate/disarm the buffered printing of Ramptr.

Campionato [Championship] 2.00: (CAMP200.ZIP) Author: Andrea Carpi. Program for filing the results of games of the Italian soccer championship and printing and displaying the classifications. In Italian, it works with Pointer Environment and Menu Extensions. The package also includes the original version of the program (all by Andrea Carpi) for Zx Spectrum, in Z80 and ZTA format.

Charselect 1.00 (CHARS100.ZIP) Author: Andrea Carpi. Utility that utilises the CHAR_SEL function of the Menu Extension for choosing a character and putting it in the stuffer buffer. Useful for finding those characters that are not on the keyboard.

Config Plus 1.00 (CONF100.ZIP) Author: Franco Aiazza. New Config_bas for the four Psion programs package. It works with Pointer Environment and Menu Extensions.

The Reader 0.40 freeware (READ040.ZIP) Author: Franco Aiazza. Off-Line Reader QWK that works under Pointer Environment. It requires Turbo Toolkit and Menu Extensions. Last freeware version.

The Reader 1.23 shareware (READ123.ZIP) Author: Franco Aiazza. Shareware version of The Reader. Much more complete.

Screen Viewer 2.18 (VIEW217.ZIP) Author: Andrea Carpi. Screen and _pic images viewer (in QL, QXL and Aurora format) that works with Pointer Environment and Menu Extensions (from version 7.xx). It can also work as a wallpaper and slideshow program. It uses File Info II if present. It can also convert the images from screen format to _pic and vice-versa. An earlier version (v2.10) was reviewed in QL Today recently.

CD PLayer 1.02 (CDPLA102.ZIP) Author: Andrea Carpi CD. Player for QPC (reviewed in a recent issue of QL Today). It uses Pointer Environment.

All the programs that work with Pointer Environment have been made possible thank to Albin Hessler's Easyptr system.

All the programs are available for the File-download requests on BC BBS (2:334/210 tel. +39-161-258677), QItaly BBS (2:331/123) and Ergon BBS (2:332/534). Some old versions can be found on some FTP sites.

If you want to contact the Beginners' Club, you can do so by snail-mail to the following address:

Beginners' Club
Via Crispi 20
13100 VERCELLI VC
ITALY

Or by Matrix to: Franco Aiazza: 2:334/210
Andrea Carpi: 2:334/210.1

Or by E-Mail to Paolo and Andrea Carpi:
carpipea@email.s-edp.it

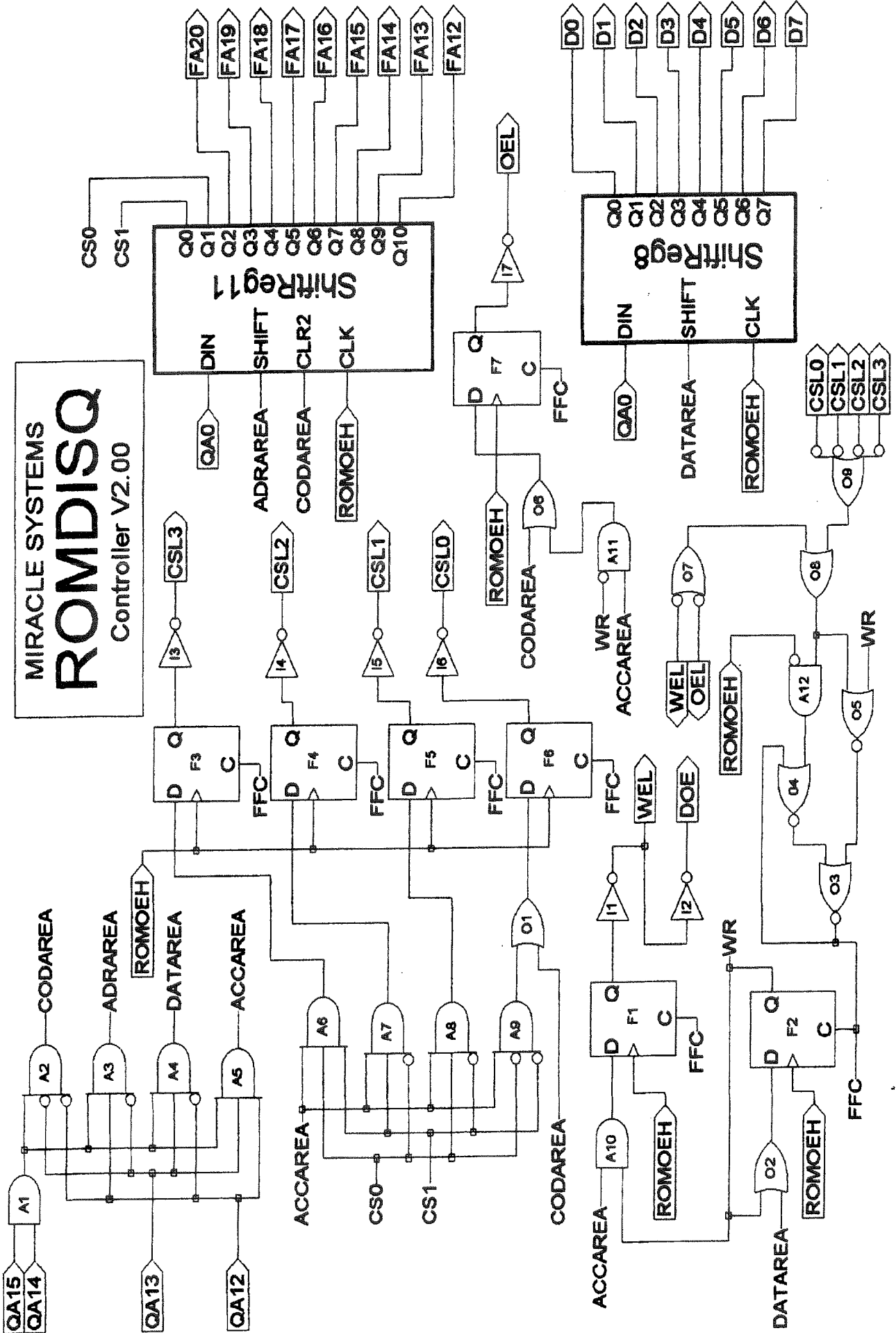
Now you can also find the BC on the Web at
<http://www.geocities.com/SiliconValley/Lab/5011>
even if only in Italian at the moment.

Logical Fundamentals - Part 4

The ROMDISQ Logic *Stuart Honeyball*

The first 3 articles in this series should have supplied enough grounding in logic circuitry to now appreciate a real product. The Romdisq is the latest product for the QL and is also relatively simple. It consists of some flash memory chips and a controller chip. The flash memories store the driver code and the user data and the controller circuit interfaces these to the QL via the QL's Rom port. Here we will concentrate on the controller since this contains all the logic.

View the controller circuit diagram reproduced here (next page). Note that the AND gates are labelled "Ax", OR gates "Ox", inverters "Ix" and flip-flops "Fx". All 7 flip-flops shown in the circuit each have the normal D and Q signals, the signal with the chevron is the clock input and the C is an asynchronous clear input. There are 2 shift registers: ShiftReg8 has 8 bits and shifts on the CLK rising edge if the SHIFT input is high; the ShiftReg11 has 11 bits and in addition bits 2 to 10 are synchronously (i.e. on a CLK rising edge) cleared when CLR2 is high leaving bits 0 and 1 unchanged. On both of these shift registers if the shift and clear inputs are low then the outputs remain unchanged after the clock edges. Signal names enclosed in arrow boxes are input and output pins with the exception of DOE which is a three-state control for D0..7 (D0 to D7). You can see that there are more than 1 arrow boxes with the same name (e.g. ROMOEH occurs 7 times) which means

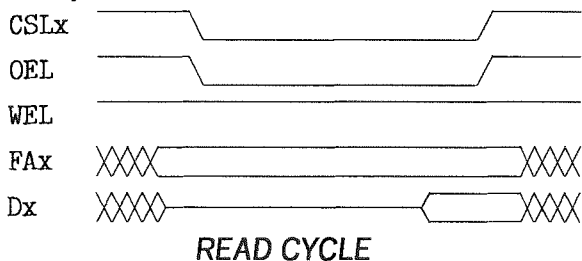


they are all the same signal. This is an alternative to connecting up all the instances of this signal with a line.

The signals QA0, QA12..QA15, ROMOEH come from the QL's ROM Port; FA12..20 are the 9 flash memory addresses 12..20; OEL and WEL are the flash memory write and read controls; CSL0..3 selects each of the 4 flash memories; and D0..7 connects to both the QL ROM Port and the flash memory data buses. The signals whose names end in 'L' are active low. The remaining 12 flash memory addresses 0..11 come straight from the QL's ROM Port.

The purpose of the controller is to increase the number of address lines and to make write cycles available. The ROM Port normally allows for only 14 useful address lines (giving a 16K byte address space) whereas the flash memory chips are each 2M bytes requiring 21 address lines and there are 4 of them effectively requiring a further 2 address lines. It is also necessary to use part of flash memory to store the QDOS device driver code to link in the ROMDISQ to the QL operating system. On other peripheral interface cards, like the GOLD CARD etc., there is a separate ROM to hold the driver code but on the ROMDISQ the fact that it is a storage device using a type of ROM means a certain economy is available.

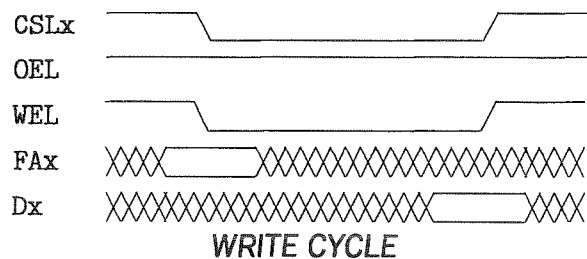
So what does it do? It is firstly necessary to know how the flash memories are accessed. They require to be read from and written to. A read cycle looks like:



The read cycle works as follows. The address lines FAx are set up (some will be high and some low) prior to the falling edges of CSL (chip select active low) and OEL (output enable active low). Just 1 of the 4 CSL signals goes low selecting 1 of the 4 flash chips. After the CSL and OEL have gone low the read cycle has started. A short time later, known as the access time, the data has been retrieved from the memory array deep inside the chip and is made available on the 8 data lines D0..7. Note that these data outputs connect directly to the QL's data bus; in other words they are electrically connected directly to the data outputs of the

other 3 flash chips, the D0..7 of the ROMDISQ controller, the Motorola 68008 processor inside the QL and everything else connected to the QL's data bus. As you know you cannot normally connect 2 or more outputs together because if one was driving high and the other low then what level would the line have? The way round this is to be able to switch outputs off so they are neither driving high nor low and as long as only one is switched on at a time there is no ambiguity. Here, just one of the flash memories is driving 8 data bus lines D0..7. At the end of the read cycle the data is read into the QL and CSL and OEL are taken high completing it.

Write cycles are used to send information to the memory and look like:



As you can guess from the write cycle waveform diagram the address is clocked into the flash memory chip on the leading edge of the cycle when CSL and WEL go low. This time data is travelling to the flash chip which clocks it in on the trailing edge of the cycle. This data comes from the ShiftReg8 in the controller which drives the data bus D0..7 during the write cycle.

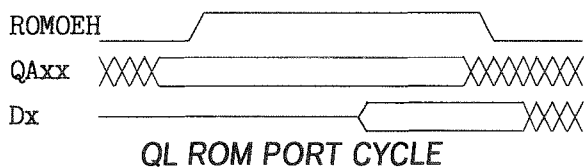
The method of increasing the address range of the ROM port and allowing for write cycles is to divide the 16K byte address space of the ROM port (48K to 64K of the QL address space) into 4 areas each of 4K. The first 4K (48K..52K) is reserved for the device driver code and is referred to as CODAREA on the logic diagram. It is essential that this code is put here because QDOS looks for it only at the start of the ROM port address space. The 2nd 4K (52K..56k) is the address area (ADRAREA) and is used for setting up the upper 9 flash chip addresses (FA12..20) and selecting which flash chip is to be targeted. The 3rd 4K (56K..60K, DATAREA) is similarly used to set up the data in the case of a write cycle and the 4th (60K..64K, ACCAREA) is for executing the read and write access cycles from or to the flash chips.

Now for a tour of the ROMDISQ logic controller circuit. In the top left of the diagram is the address decoder. The QL's ROM port

actually spans a 64K range of which only the top 16K may be used. To extract this 16K the QL address lines QA14 and QA15 must both be high so they feed gate A1 - any other combination of values is outside of this 16K. QA12 and QA13 are used to select each of the 4 areas of 4K using gates A2 to A5.

Below the address decoder is another decoder. The 2-bit binary number on CS1,0 is decoded by gates A6 to A9. The signals then go through flip-flops F3 to F6 and inverters I3 to I6 to give the 4 chip selects CSL0..3. Only one of these can be low (active) at any time and only when an access is taking place. Notice the extra OR gate O1 which allows access to the code area bypassing the general accessing regime. It is clear that the driver code must reside in flash memory chip number 0.

ShiftReg11, like all the other synchronous elements of this circuit, is clocked by ROMOEH. The QL can only read from its ROM port and a cycle looks like:



The rising edge of ROMOEH starts the cycle off and the ROMDISQ controller clocks in all address related information it needs at this point because the address lines are not guaranteed to be valid right through to the end of the cycle. ShiftReg11 gets its data from QA0 so when the QL reads address 52K+0 then a 0 is shifted in and when it reads address 52K+1 then a 1 is shifted in. To fully load up the shift register 11 reads are necessary. (Note that the actual data that the QL reads back during the loading up of ShiftReg11 is meaningless and is discarded.) The value for FA12 is shifted in first although initially will appear on Q1 and so on until it appears on Q10 where it is required. However, if at any time an access to the code area is made then Q2..10 are all set to zero. (The reason for this is to make the software more efficient and is outside the scope of this article.) ROMOEH can also be toggled when neither ADRAREA nor CODAREA are high and will have no effect leaving the state of ShiftReg11 unchanged.

ShiftReg8 is a simpler version of ShiftReg11. The data area is read to load up this shift register. When the QL reads 56K+0 the value 0 is shifted in and reading 56K+1 shifts in a 1. D7 is shifted in first and makes its way down to Q7

during subsequent shifts while the rest of the data is shifted in bit by bit.

Down in the bottom left of the diagram is flip-flop F2. This is set high whenever the data area is read. It signifies that the next flash memory access is to be a write. It is cleared immediately after the access has taken place. The OR gate O2 makes sure that ROMOEH pulses happening in between times do not cause it to clear.

After the 2 shift registers have been set up an access cycle can now take place. It is either a read or write depending on WR driven by F2 which would only have been set if the data area had been read. For a read cycle ShiftReg8 is not loaded because it is not needed and so the Q of F2 (WR) will have remained low. To perform the flash memory access cycle the QL reads address 60K+LowAdr where LowAdr is the bottom 12 bits (A0..11) of the address to be accessed. It is important to realise that, although the QL does a read cycle, the flash memory will 'feel' a read or a write depending on whether the data area was read during the set up. In the case of a write the output enable DOE will be high and will enable ShiftReg8 to drive D0..7 (this mechanism is not shown here). This data output enable, DOE, is driven by an inverter, I2, from WEL rather than directly from F1's Q for reasons of timing. The D0..7 must be driven until WEL has gone high so DOE must lag behind WEL.

At the end of a code access or a general read/write access to the flash memory, on the falling edge of ROMOEH flip-flops F1..F7 are all asynchronously cleared. There is an additional flip-flop comprising the NOR gates O3 and O4 down the bottom of the diagram which takes part in this clearing by driving FFC connected to the 7 flip-flop clears. The gates O7,O8,O9 are really just a 6 input OR gate with each input inverted. When any of these inputs are active then O8's output is high. At the end of a cycle ROMOEH will go low and so A12's output will go high. This will set the O3,O4 flip-flop output (FFC) high and will start all the flip-flops clearing. When they have all cleared then the NOR gate O5's output will go high, reset the O3,O4 flip-flop and the clearing is complete.

I hope this has gone some way to explaining how a real world logic circuit works. The only real way of understanding the circuit fully is to go through it yourself following what happens during various QL read cycles. If you feel that any of it is more complicated than is necessary then try your simpler logic and see what

happens. Part of the reason for doing the circuit in this way was to fit it into a Lattice 2032 programmable logic chip which you will find on the ROMDISQ with exactly this circuit inside it.

The future of ProWesS - Part 1

Joachim Van der Auwera

On the QL-user mailing group, there has recently been a large discussion about the development of device drivers on the QL. This discussion included an appeal to have some discussion when developing OS extensions or modifications. As a result of that, I distributed some details about plans for changes in the ProWesS window manager. Jochen Merz asked me to pour this into an article about the future of ProWesS, for publication in QL Today. That's what you are reading now.

First some clarifications. The ideas about the future of ProWesS are not cast in stone. Though some of these improvements are being worked

on at the moment, changes are still possible. I would like to get as many comments as possible about these ideas (both what is good and bad, and definitely about omissions etc). This can only help in making ProWesS better, and making the QL a better and more powerful environment. Also, if some of you would want to contribute in some way in the development of ProWesS, you are more than welcome !

The original design info which was distributed only discussed the changes in the ProWesS window manager. Any other part of the ProWesS package was not discussed. In this article, I will try to make up for that, and will include my ideas for the future of the entire ProWesS package, and some ideas about the future of the QL OS (which I hope ProWesS will be an important part of).

The ProWesS window manager

The ProWesS window manager currently has a few limitations which can be very annoying when writing certain types of programs. There are a few things which are currently not supported, but which could make life much easier, or could make ProWesS more attractive and/or user friendly.

WE SUPPORT SINCLAIR

QBOX USA

COMPUTER BULLETIN BOARD SERVICE
(810)254-9878

**We support *all* SINCLAIR COMPUTERS
(QL, SPECTRUM, ZX81, Z88, Thor, QXL)**

- Now in our 4th year on-line round the clock since October, 1993
- Full message area and File Download areas
- We carry all popular SINCLAIR message areas from Europe
- Calls from 14.4k--300 baud are welcome
- QBOX - USA runs on a SINCLAIR QL with Super Gold Card, Hermes, QUBIDE, 200MB drive, USR sportster 14.4 modem

→ NO FEES CALL US

One of the things which can be improved is the efficiency and power of object types, and better handling of messages and events.

Difference between an event and a message:

An object can react both to messages and events. In principle events are generated by ProWesS itself and messages can also be sent by the user using the PWChange command).

The difference on the implementation level is that an event is handled at the object level - thus may be different for each object - while a message is handled by the type. Also messages are forwarded to the parent object when an object doesn't handle it itself. This passing of unhandled messages to the parent object was not done in ProWesS v1. For queries the query is normally passed on to the parent object when not answered, but it can also be flagged that only the designated object can get a chance to answer the query.

Message handling: When handling messages using the PWChange keyword, the system should first ask the recipient object to handle the message and if this object does not wish to handle the event, the message is passed to the parent object to allow that to handle it. This continues to the grandparent object etc. To make this process slightly more efficient, the object type can indicate mask of messages it may accept. This can allow the PWChange routine to know in advance if it can skip an object as it will never accept the message. The messages will be handled by separate routines for each message. This allows the compiler to do more and better optimizations. Each message routine gets two objects as parameter: the object which receives the message, and the object to which the message was passed.

Event handling: It would be better to have specific event handling routines for each possible event. This is more efficient, and gives cleaner code. To reduce the memory consequences, there are general handlers for the type and handlers for the specific object. The object is searched for the event handler before the type.

These indicate some changes in the internal mechanisms which are used in ProWesS and which are important when writing ProWesS types. Most of the improvements are direct consequences of the limitations of the current system.

When positioning objects in ProWesS v1, objects are positioned in rows and columns. The design is flawed as rows can actually change to columns and vice versa, so this will be fixed. Also the current version allows objects in a con-

tainer to have the same height (in a row) or identical width (in a column). More powerful constructs are currently not possible. This is now remedied by having containers which actually determine the size and position of the objects in it (though the object can tell its preferred size). This means it will be possible to have objects in a grid all with identical size. Another feature which is not possible in ProWesS v1 is a "multiple document interface". This means that the objects in it can overlap. It will also be possible to have a container which is smaller than the contained object. Using this kind of object, you can introduce an object to include some other object to make them smaller and scrollable.

Improved positioning: All the objects which can be created should be positioned in an object tree. Each system contains one object tree, and this tree represents all the menu objects in window which can represent that system. The object types can be of two types: branch objects or leaf objects. A branch object can have both branches and leaves connected to it, but a leaf object can not have any children in the tree. Visibly speaking, all objects are always fully contained within the parent objects area. Though the virtual area may be bigger, the visible part can't be. Objects which are at the same level in the object tree are contained in the same parent objects and may (partly) overlap. The objects are ordered from front to back. To make window redrawing efficient, the parent object has to indicate whether it is possible that the children overlap.

Size of an object: The size can be determined by each object individually. The object can determine its preferred minimum size. Normally speaking an object will make sure that the contained objects are completely visible (except for things like a "multiple document interface"). Each object has to assume that its size can be changed later on. It can be notified of this (to allow resizing of child objects or making sure the internal state remains correct). This is because the parent may resize the objects to conform to some kind of rule (e.g. all equal width). In principle, a window has a fixed size (which may be scaled), but some objects may tell the system that they want to resize depending on the amount of information in them and the amount of screen space. These objects can then resize ("autosize") their contents (they should know the possible increase amount). When there is room left, the next object which allows autosize will get a chance.

Keypresses will be handled similarly to

ProWesS v1. There will be some differences. Normally keypresses are handled by the system, only if there is no action for the keypress, then it is passed on to the "current catch object". Switching between catch objects was quite cumbersome to set up, you had to tell each object what the next and previous one is, which is easily bugged, and very unpractical. This will be improved. For the keypresses which are directly defined, it will be possible to check which keys are already in use. This allows automatic assignment of keypresses by the objects!

Keypress handling: All keypresses should be handled by the system (this is also true in v1, although this was not stated as clearly). It should be possible to query whether a keypress is already "in use". Each keypress can be connected to an object. When the keypress is activated, a user defined (e.g. "keystroke") event is sent to that object. Obviously several keypresses can be attached to one object. However, each keypress can only be linked to one object. Keypresses are case sensitive. However, when a key is pressed which is not linked to an object, then the case is changed and a matching object is searched again.

Switching catch objects: All objects (or their type) have to contain a flag which indicates whether it is possible to catch keypresses. This doesn't mean this is always the case, but just that it could be possible. Objects could then be stored in a linked list of "possible catch objects" when they are added to the system. The order in this list is then the default order for switching catch object with <tab> or <shift tab>. Of course you have to be able to explicitly change the order of objects in the list. When moving through the list, it should still be checked whether the object is effectively willing to accept catch events.

When objects are deleted in the old ProWesS, all the child objects are also deleted. In ProWesS v2, it will also be possible to delete an object, while keeping the children in the menu. This can be useful when an object has reduced its size so that you no longer need a parent object which makes it scrollable.

Deleting objects: To make sure that objects can contain links to other objects and make sure their internal state remains consistent, all objects in a systems need to be notified when an object is removed from the system.

Some other important changes are the handling of move and resize of a window. In the current version, each window automatically get a scaleborder to allow this. In ProWesS v2, there

will be no default handling, all move and scale events have to be performed by an object (normally as a result of a user action).

Creating a system: The handling of move and rescale should be initialised by the types. The system should not incorporate how this is interfaced to the user. There should be a type which is intended to be the top level. When this is made compatible with v1, the type will implement the scaleborder. When a type is created in a new system, then the system has to include the top level itself (if the object was not of that type). In principle this should not be necessary, but it is done for extra user friendliness and for compatibility with v1. This option can be disabled by passing a creation tag.

Window scaling: It will be possible for a window to be bigger than the area which is used on screen. This can be because the window is larger than the screen, but may also be forced otherwise. As a result, you could allow the user to shrink the window.

There are some smaller changes. The window will be redrawn more often to inform the user more quickly. This means that you no longer have to do that in your types except for some special cases (for efficient sub-object redraw).

Redraw of objects: The system should redraw more often. In particular, the window should redraw at the end of each access to the system if the system is active and does not need resize.

There are some important changes in data hiding. The internal structures will be better hidden from the types. This means that in the future, the system could be modified to allow sending messages to objects in a different job, without the need to change the interface for the user or the types.

Data abstraction: All the internal data structures should be hidden. Each routine should get at least two parameters: the object identifier (used to interface with the ProWesS system) and the ObjectData structure. This structure can be defined by the type.

Multitasking options: The ProWesS interface should be defined in such a way that it would be possible to access a ProWesS object from a different job then where it was created. This means that there should be no direct access to memory except in some well defined cases (e.g. the ObjectData). Definitely no access via object id's which happen to be pointers (very deadly). The multitasking stuff will not yet be implemented, as I fear that this may slow the system down.

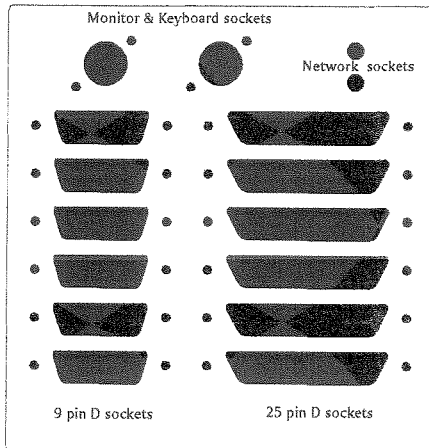
Some implementation related changes are the

Hardware

QXL II	£ 200.00
Super Gold Card	£ 160.00
Recycled Gold Card	£ 60.00 *
Aurora	£ 100.00
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
The 'Braquet'	£ 16.00

* when available.

New Product ! The MC - Plate



All the socket holes you will ever need on the back of your tower case. Just rip off the silly PC thing and screw on this one. No cutting needed. What could be easier ?

Package deals

QXL II + SMSQ/E	£ 260.00
Aurora + SMSQ/E	£ 160.00
Aurora + SMSQ/E + Super Gold Card	£ 330.00
Above package + Monitor	£ 380.00

New Program !



Hard Disk Backup Program
How much is the data on your hard disk worth ?
How safe is it ?

£ 30.00

Other Q Branch Programs

Q - Count

£ 25.00

Q - Route v1.05

£ 25.00

Page Designer 3 S

£ 40.00

Q Branch

Feeling out on a limb ?

Reach out for Q Branch

Suppliers of Quality QDOS/SMSQ products
Hardware and Software



P.O. Box 7.
Portslade
E. Sussex

BN41 2ND

Tel : 01273-386030

Fax : 01273-381577

Email :

qbranch@qbranch.demon.co.uk

Q Branch have, once again, been increasing the size of their catalogue in the last two months.

New releases at this time include a steel backplate which can be fixed to most mini tower cases to give solid mounting fixtures for serial, parallel, keyboard, network and monitor outputs.

We would also like to announce the imminent release of the SBASIC / SuperBAISC reference manual. This is the complete guide to all of the keywords that can be found in QDOS / SMSQ/E as well as the many toolkits that are available. The manual also gives tips on programming and cross references to other related keywords as well as telling the user where he can find these extensions.

On the software side we have a new release of Solvit Plus 3 which now has a pointer driven front end and an upgrade to the Thesaurus. Mark Knight is putting the finishing touches to the new version of The Knight Safe which will compress the files on the backup device decompress them when restoring. We hope that we will soon have this and Barry Ansell's new music program in the catalogue.

Call for details.

We are looking into some new hardware devices and hope to have news of some of these at the various workshops during the coming year.

Why not come and talk to Q ?



We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.



Q Branch

Feeling out on a limb ?

Reach out for Q Branch
Suppliers of Quality QDOS/SMSQ products
Hardware and Software

P.O. Box 7, Portslade, East Sussex.
BN41 2ND
tel: 01273-386030 fax: 01273-381577
Email :
qbranch@qbranch.demon.co.uk

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK, SOLVITPLUS 3

£ 15.00 ea / ANY 2 PROGRAMS £ 25.00 / ALL 3 PROGRAMS £ 35.00

QPC v 1.42

SMSQ/E owners

£ 70.00

non SMSQ/E owners

£ 90.00

PROGRAMMING

QD v 9.04	£ 53.00
QD + QBasic	£ 69.00
QD + Qliberator + QBasic	£ 110.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 37.00
Easyptr pt 1	£ 37.00
Easyptr pts 2 & 3	£ 18.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 37.00
QMenu	£ 16.00

!!! NEW VERSION !!!

THESAURUS V4.01

Pointer driven version of Geoff Wicks popular program.
(needs Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)
Still the same price

£ 15.00

Upgrades from v 3.xx £ 2.50 + S.A.E.

SMSQ/E

Gold Card / Atari / QXL version

£ 76.00

Various Atari versions : call for details

UTILITIES

FiFi 2	£ 22.00
QSup	£ 32.00
QSpread	£ 66.00
Cueshell 2	£ 37.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 32.00
LDUMP	£ 26.00
Q Count	£ 25.00

Q Branch high quality Mouse Mats

£ 5.00

ProWesS

ProWesS	£ 48.00
DATAdesign	£ 24.00
Fontutils	£ 30.00
File Search	£ 12.00
PFlist	£ 12.00
Fontpack	£ 60.00
LINEdesign v 2.16	£ 24.00
PWfile	£ 18.00

Text 87

£ 79.00

Typset 94 £ 29.00
Fountext 94 £ 39.00
2488 drivers £ 29.00
Epson ESC/P2 drivers
£ 26.00

Text 87 is the only QDOS /
SMSQ wordprocessor
capable of handling the full
screen on the Aurora / QXL /
QPC systems. New drivers
are currently being written.
WATCH THIS SPACE !

You've been asking for it and here it is !

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including
three disks of PD toolkits, example procedures and an electronic index.

compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500
pages !

Call for
pricing details

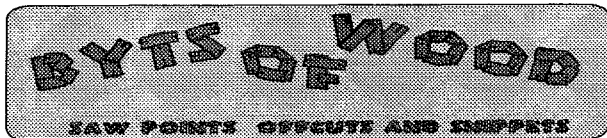
Check out our New web site for all the latest news
<http://www.qbranch.demon.co.uk>

deprecated PWAddType call (I don't think anybody used it up until now, a dummy will be implemented which just returns an error). The handling of unknown messages is also handled. It can be annoying to return an error for an unrecognized message (e.g. because the type is not the very latest version), as the added fuctinality may not be vital for the programs functioning.

Unknown messages: The handling of unknown messages has changed. In v1 you would get an ERR_IPAR after handling the remaining messages. This is now different. If a message is not recognized by the object, its parents or the system, then the tag is just skipped. For debugging purposes however, you can query the system to know how many messages in the last PWCreate or PWChange were not handled.

Of course, when there are going to be so many changes, there is always the problem of backward copatibility. I realise this is very important (I would prefer not having to bother, but well...). Everything possible will be done to make sure that applications which work in ProWesS v1 also work in ProWesS v2. The types however will not be compatible. The internal workings are so different that it would not be wise to try and support the same structures. As an aside, the new structure for the types will be easier to use, so writing new types will be more straightforward. What's more reuse of types will also be much easier.

■



Driftwood in the Driftnet

At the start of January I transcribed and did a precis of a long electronic conversation that went on over the ql-users internet conference. (See Netted - this issue). In the course of this transcription a few things occurred to me. Some of these I injected into the text as I transcribed it but there are a few points which I would like to expand here. Over the past fifteen years we have become a community of people who are linked in only one way and that is the use (or abuse) of a particular computer system. The attrition of time and the needs of the general user have caused the people who remain die-hard QLers to be a very assorted bunch and the polarisation of that collection of individuals can be seen very clearly if you have to deal with them as Jochen, Ron Dunnett, myself, etc. do. If you read through the opposing messages put

out mainly by Nasta and Thierry Godefroy you can see the hardware designer versus the software writer in great detail. Nasta wants to create the perfect hardware platform and has a very definite vision of what that platform should be whereas Thierry wants to perform certain tasks with as much elegance of code as possible. Both these approaches are laudable but they do lead to a certain dichotomy when applied to the practical world. The end-user is impatient and cannot wait for perfection (the ideal not the in-aptly named program) and so will begin to look around. When it comes down to the bottom line in things the computer is a tool and not a team sport. You may support the QL and all it stands for but, if it cannot do what you want it to do, it is, in the end, useless. Like one of the correspondents in the conference I do 99% of my work on the QL but I also have a PC which I use for internet access. The very fact that I have and use a PC does not disqualify me for anything but it does, as John Hall pointed out to me once, expose me to a different system. I do enjoy using the QL and that will not change, I hope, but I do also look at the way a PC does things and, in making that comparison, learn things about both systems.

CONF_USE

Certain things came up during that discussion that I find a source of constant irritation. The first is the DATA_USE/PROG_USE situation. I prefer my hard disk to be subdivided into compartments which neatly label the contents. Some programs force me into sticking things into those compartments which do not belong there. I have to put a certain file in the DATA_USE root directory because the programmer has decided to use that area to store his data or because he has decided to get his program to look there first without giving me that choice. I could, of course use DEV_USE or Phil Borman's PATH utility but in the first case I am interfering with the way in which the operating system runs and the second I am adding yet another extension onto the system and, until the Gold Fire arrives, space and memory are both tight. Recent discussions on the above mentioned internet pages have centred around the 'problem' of QDOS/SMSQ file names and the fact that they also include their paths. As a user I like this way of doing things because I can see where each file is going but maybe serious programmers have problems with this. Of course the situation on my PC with Windows for Workgroups 3.1 is even worse. I have only to save to a file on floppy once in a session and the dammed thing looks

there every time it has to do anything. Windows 98, apparently, has an even more annoying habit. If it cannot find a file it is looking for it pops up a message saying 'I cannot find that file - shall I look on the internet?' There are ways around this. I have no problems with Config blocks whatsoever and the level two config makes upgrading simple whilst leaving me in total control of my system. Jonathan Hudson dislikes the config block system and opts for Environment Variables which (as he has pointed them out to me in detail) are also simple to use. So this is a plea to the programmers out there - 'Forget PROG_USE and DATA_USE' and give control back to the user.

Its Hard but not Disk-usting

One other thing that I have often complained about is the way that the user is unable to make a subdirectory on a Qubide hard disk if a file with that name as part of its path already exists. This is a DOS standard but we were able to do this on all other QDOS/SMSQ systems before. I got many responses to this complaint including the industry standard, 'You're the only one who complained about this' but when the above discussion on file name length and paths came up I threw it into the ring again to see who would respond. Lo and Behold, not only did a few more people say they hated it but Phil Borman, who wrote the code, finally explained that the space restraint on the Qubide ROM meant that he was unable to fit the extra code needed into the 12k provided. It was a choice between support for multiple drives and multiple partitions and this feature. So now I know and I will shut up about it.

The Old School Ties

In the course of January I came across some of the problems which regularly beset the software traders. I sold a copy of QPC to someone so that he could use it on his laptop. He came over to the Q Branch HQ and I did the installation for him. I wrote him a boot file and installed a basic QPAC 2 setup as well. He went away but a few days later he was back. The problem here was that he was working for a company who did all of their work, invoicing, etc in Archive and he, having never heard of the QL until then, had to get his head around a whole new concept. I installed Xchange on his system for him and did not think any more about it but Xchange relies on those PROG_USE/DATA_USE defaults and, although it will run fine without them cannot find the help files or the printer data if they are not set up. I have not used this system for some

time (since 1988 in fact) so I was completely confused for a whole day while I tried to sort it out. The boss of the company had written all of the company's programs in the Archive programming language without any reference to the drives that the data was on. He only used a Trump card and twin DD drives so he never had to do this. Here we have a modern emulator trying to run a fifteen year old piece of software and the response from the end user is not 'ah I have to put some work in to adapt my way of thinking here', but 'The emulator does not work'. Sir Clive, in his infinite wisdom, did not even think that floppy disks were worth considering so a hard disk was out of the question and here we were trying to get software written for that era to perform the same on a system with both of these.

Everyone's A Win-er

Another area where people get confused and find themselves in need of help is the use of multiple 'win' files. The Qubide offers you software to construct several different virtual hard disks. On the disk provided by Qubbesoft there is a suite of different pieces of software by Phil Borman. By using these programs you can create 'win1_', 'win2_', 'win3_' etc and link these in and out at will. People used to this system want to create similar files on the QXL or QPC but there are differences in the approach to this which need to be explored in depth. There are a few things that need to be clarified, especially for the new user.

1. 'Don't be afraid'. Typing 'Format win1_' from QPC or the QXL will not format the entire hard disk. What it is doing is creating a file of a fixed length called QXL.WIN on the device that DOS calls C:. If you do a directory from DOS you will see it there.

2. Do not make a subdirectory called QXL.WIN and expect that to be recognised by QPC/QXL - it won't it is only a DOS subdirectory and it will stop the emulators creating the real one.

3. If you have just bought a QXL read the README file on the disk. Tony Tebby found that some people accidentally formatted their hard disks so he built a security device into the 'format WIN1_' command. You have to type the following:

```
WIN_FORMAT 1
FORMAT WIN1_xxx
```

(where xxx is the size in Mbytes that you want for a hard disk) You will then see two letters appear on the screen. Type these in in the case that they appear and the formatting will start. Now we need a win2_..... The method of creating

WIN2_, WIN3_ etc on the QXL id different from that used on the QPC emulator so I will tell you how to do each one separately.

The QXL way

The QXL can 'see' other storage devices attached to the PC if they have a QXL.WIN file on them. This means that a CD-ROM that has a QXL.WIN file in the correct format will be recognised by a QXL if you look in the right place. The naming regime of DOS computers is the the first floppy drive is called 'A:\'. If you have a second floppy drive that is called 'B:\' and the hard drive is called 'C:\'. From then on it is anyones guess and all the other addresses are up for grabs by whichever device comes along first and stakes a claim. That does not concern us too much here. QXL sees the drives in the following way:

```
A:\ = flp1_  
B:\ = flp2_  
C:\ = win1_
```

When we type 'format win1_' inside the SMSQ or SMSQ/E system on a QXL we effectively create a large DOS file called 'QXL.WIN' on the hard drive. Now we have a choice. If we have a second hard drive connected to the PC which the PC sees as 'D:\' we can 'format win2_' in a similar manner to that descibed above and get a QXL win file on that drive too. This will also work for an EAZYDRIVE or TAPE DRIVE system connected to the PC as 'D:\'. If we want a 'Win2_' device but have no second drive we have to get involved in a bit of DOSery. First look in the CONFIG.SYS file and see if there is a LASTDRIVE= command there. IF there is and it says

```
LASTDRIVE=D  
change that to  
LASTDRIVE=E
```

Then go to DOS or the File Manager from Windoze and create a subdirectory. Call it whatever you like - for our example I will call it 'C:\WIN2\'. Now go to the AUTOEXEC.BAT file and insert the line:

```
SUBST D: C:\WIN2\
```

Save the file. When this is done reset the computer and, if you look in the File Manager you should see a device called 'D' which is just your 'WIN2' subdirectory in disguise. Now go to SMSQ\E) and type the magic format win2_ command. You now have a Win2_ device. You should, of course be careful about how you allocate these devices. If you have a device called D:\ alreday there is no way that you can use SUBST to create another one. You should therefore create the next device up in the list. Be

aware also that 'win2_' is device 'D:\' and if you have found that you have to create a device called 'F:\' to get it to be the next in line you will have to type 'FORMAT win4_' to get the extra partition.

The QPC way

QPC is different. When you configure QPC you will see a number of options in the configure menu. 'name for win1_', name for 'win2_' etc. They are mostly called C:\QXL.WIN when you start but if you change win2_ to, say, C:\QXL2.win you can create a new section on the drive. In order to create a win2_ device all you have to do is to format that device and you will see in the appropriate directory a file called QXL2.WIN. If you have a second hard drive installed as, say 'E:\' then change the config item to E:\QXL.WIN and the sub-directory will appear there as QXL.WIN. This is a much more flexible system than that used by the QXL. If you upgrade your copy of QPC you will have to restore this configuration but there should be a program on your disk called Qconfig which will save your configuration and restore it when you do your upgrade - simple or what ?

C-dese

Using these two methods you can have as many 'win' drives as you want. I do not know about writable CD roms however. I believe that it should be possible to copy a preconstructed QXL.WIN file onto a CD-ROM and then have the QXL or QPC read it back but it may not be viable to write that data to the CD-ROM in bits. If it has any similarity to Audio CD it must keep track of its data in a single section so how this could be achieved I am not sure. Although the format command creates the QXL.WIN file of a fixed length I do not think that the CD driver has no mechanism for writing to that file. If anyone has any experience of any of this please tell me. We do plan to conduct a few experiments here at the Q Branch HQ to find out. I will let you have the results in the next issue.

DOS and other swear words

A recent purchaser of QPC reported to me that, after he installed QPC, his computer would not run windows anymore. (This seemed logical to me because, having seen a more efficient system at work, WINDOZE then decided it could not compete and gave up but I did not really believe this). I installed QPC onto my new laptop without any problem whatsoever but that was not the end of the story. The strange thing about this

Sinclair QL
PD
and
Shareware
Software

QUBBESoft P/D

Sinclair QL
New
and
2nd User
Hardware

38 Brunwin Road, Rayne, Braintree, Essex. CM7 5BU
Tel/Fax: +44 (0)1376 347852

Aurora

The QL's Graphics Card

- Replacement for 13 year old QL Motherboard
- QL/VGA/SVGA or Multisynch Monitors
- Uses SMSQ/E Supported Operating System
- 8 Resolutions from 512x256 upto 1024x768
- Super & Gold Card (Red) compatible
- Future 16 & 256 Colour modes
- superHermes, Di-Ren Kybd Interface compatible



(P.R.P. £120.00p) Now Only £100.00p

Includes A4 Manual & VGA/SVGA/MSync Monitor Cable

QUBIDE

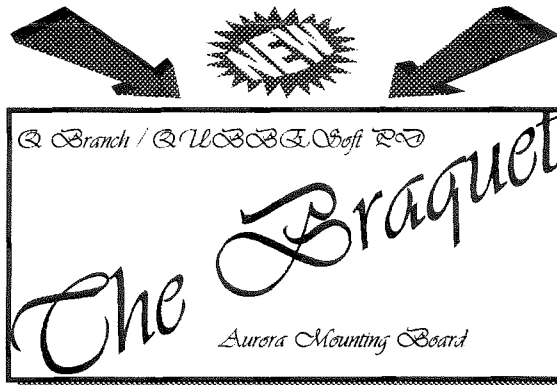
Allows AT/IDE Hard Disks to be connected to your QL/Aurora. A huge amount of storage space will become available for files and your programs Compatible with SGC/GC/TC and all major add-ons.

Only
£55.00p

QPLANE

Back Plane for use in placing your Aurora or QL into PC Mini Tower/Desktop Case. 2 Power connectors that marry up with special PC style connectors. 3 slots for plug in cards

Only
£25.00p



Only £16.00p

Includes Fixing Screws & Spacers

Floppy Disks Pre-Formatted QL

10 DS/HD
£2.50p

10 DS/DD
£3.00p

PD - LD Clip-Art - Electronic Book Library Collections

PD Collection - The complete QUBBESoft Library of PD and Shareware Software.

LineDesign Clip-Art Collection 1 & 2 - Over 100mb of LineDesign Clipart & loads of Fonts.

Electronic Book Library Collection - Classic Literature, Grammar course, SCI-FI & lots more

On EZ 135 Cart
£25.00p
Including Cartridge

On DS/HD Disks
£25.00p
Including Disks

On Iomega Zip
£20.00p
Including Disk

2nd User Hardware & Software

A large collection of 2nd User Hardware and Commercial Software available. Stock continually changing please ring for current availability. Stock list can be found on Di-Ren's Internet Webpages.

New Hardware in Pipeline

Gold Fire: Replacement for SGC, 33mhz Cold Fire Processor, Upto 128mb RAM using 72 pin SIMM's, Bi-Directional PAR Port, On board Switch Mode PSU will automatically detect input voltage and adjust output accordingly.

All prices exclude P&P please ring for details. Payments:- UK Cheque, Eurocheque, Cash. If you wish to pay by Credit Card all of the above is also available from QBranch.

phenomena is that when it happens there is really no way of telling why it happens. You can start and stop QPC as many times as you like and then suddenly - bang - no more WINDOWS. To some people that may be an intense relief but it can be very frustrating if you have to use both systems. IT is NOT QPC. It happens with the QXL-too sometimes and may even happen with other programs that have to stop WINDOZE running and start a separate DOS job. It may be part of the much vaunted 'Plug and Pray' concept that Bill Gates' crew have been trying to shove down our throats (the 'what too stupid to load and configure your software - no problem we will mess it up for you' approach).

The solution to this should be simple because when it does this it should re-label the old files AUTOEXEC.BAK and CONFIG.BAK. All you have to do is to rename the ones in the root C:\ directory back to what they should be and overwrite the other ones and everything should work fine again. This does not rewrite the files in the QPC section at all. Of course, in order to do this you have to get the machine to stop running QPC. This is done by pressing F8 as the machine is booting up and then selecting the menu item 'start with Command prompt' or 'Safe Mode (what kind of a cruel joke is that then?) Command prompt'. You can then use the DOS 'edit' command to look at the two files. I would recommend that, before installing QPC you copy the AUTOEXEC.BAT and CONFIG.SYS files under different names so you know which files to restore just in case they get completely destroyed. If all else fails, Jochen's advice is to delete the two files and start up without them because WINDOZE 95 will write them for you.

Double Glazing

The same purchaser returned QPC to me because he said 'It only works in a tiny window on the screen and that is useless'. Jochen and I have lost track of the number of times we have explained this and written about it so I have added a page to the SMSQ/E manual. Now at least no one can say it is not in the manual. While we are discussing the window situation I also have some information which will be of interest to other people who are using Aurora, QXL, SMSQ/E and QPC systems. Many people, when firing these systems up for the first time, worry like my customer above that the little red/white/black windows that they see are all that they will get. The trick here is to open up the standard QL WINDOW#0, WINDOW#1, and WINDOW#2 to the right size to fit the display size that you have selected. This is all very well and can be found in

the revision 7 version of the SMSQ/E /QPC manual but how do you know what sizes to make them and, more importantly, what sizes to change the windows in your own programs so that they will also take advantage of the new screens. John Wakefield of the Sussex User Group wrote a little calculator to do just that and I helped him to turn it into a pointer driven program. You should find this on the cover disk for this issue (screenconv.zip). Just set the size of your old screen, the size of your new screen and the size and co-ordinates of the window you used to open and it will tell you what size the new one will be. It even puts it into the stuffer buffer so you can drop it into your own programs. John Wakefield did most of the work on this so all credit to him.

Gee Graphics ! (On the QL ?) - Part 3

H. L. Schaaf

Fence Posts and Option Bases

We first learn to count from 1 to 10; later in life we zero in on the option base of 0 and concepts such as negative numbers. By now most of us have made 'fence-post' errors as we intermingle the option bases 0 and 1. Consider a line of fence to be 100 meters long with fence posts every 10 meters. How many posts are needed? Not $100/10 = \text{ten}$, but one more or eleven posts; yes there are 10 sections (or intervals or panels) of fence, each 10 meters long between adjacent posts.

Pixels and Graphic units

Pixels (picture elements) are a convenient abstract concept. A pixel value can describe attributes such as color; the location of the pixel in memory can be mapped to a particular location on the screen. For the ordinary QL we have 32768 bytes of memory starting at address 131072 (or SCR_BASE in SMSQ/E) that are used as pixels. In MODE 4 there are 4 colors, with 131072 pixels arranged in 256 lines having 512 pixels for each line. In MODE 8 there are 8 colors (and FLASH) with 65536 pairs of pixels arranged in 256 lines with each line having 256 pairs of pixels.

The BLOCK command works in pixel units, while the POINT command uses graphics units.

QUANTA



Independent QL Users Group

Worldwide Membership is by subscription only, and offers the following benefits:

Monthly Newsletter - up to 40 pages

Massive Software Library - All Free !

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

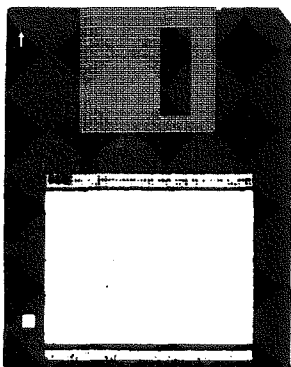
Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

*** Now in our FIFTEENTH successful year ***

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex
SS7 4JD
Tel. (01268) 754407



To put a small white block near the center of the maximum sized screen enter the following lines:

```
MODE 4 : WINDOW 512, 256, 0, 0
SCALE 100, 0, 0 : PAPER 0 : CLS
BLOCK 1, 1, 256, 128, 7
```

If I use a strong magnifying glass and take a look at the pixel represented on the screen I see a small "ball" of red, green and blue dots with almost a 3D appearance! A "globular Gaussian glow" is another way to describe it. Now plot a single point with:

```
INK 7 : POINT 50,50
```

Compare the two and then add two more with:
BLOCK 1, 1, 257, 129, 7 : POINT 51, 51

The full active display area of my RGB monitor measures 21 cm wide by 14.5 cm high. Therefore in MODE 4 I have $512/210 = 2.44$ pixels/mm horizontally and $256/145 = 1.76$ pixels/mm vertically. To put it another way a pixel is represented by an area .41 mm wide and .57 mm high on my monitor. On my monitor the individual dots are about 0.2mm. in diameter and the dot pitch is probably 0.33mm (distance to next dot, no matter what color). In terms of dots/pixel I have $1/(2.44 * 0.33) = 1.2$ across and $1/(1.76 * 0.33) = 1.7$ down. A triad of the three different colors approximates a triangle about .5 mm wide by .5 mm high. That compares well with the .41 by .57 display area for a pixel or enough to make a nice display. Just think, I can figure this all out again when I get a bigger monitor with higher resolution and the Aurora!

Aspect ratio

Aspect Ratio = width/height for a rectangle

Some terms in common use are:

"Portrait" aspect ratio less than one
"Square" aspect ratio equals one
"Landscape" aspect ratio greater than one

Golden ratio:

as a Portrait = 0.61803398874989484820

as a Landscape = 1.61803398874989484820

Aspect ratios for a few common paper sizes :

Size of sheet of paper	Orientation	
	Portrait	Landscape
A4 paper 210 by 297 mm	0.707	1.414
A5 paper 148 by 210 mm	0.705	1.419
8 1/2 by 11 inches	0.773	1.294
'Legal' 8 1/2 by 14 inches	0.607	1.647

Television and early monitors had an aspect ratio of 4:3 (or 1.333) which could be adjusted slightly for vertical size and horizontal width.

Human binocular vision sees a region 200 degrees wide by 135 degrees high for an aspect ratio of about 1.48 and the central overlap region seen by both eyes has an aspect ratio of about 0.9 being 120 degrees wide by 135 degrees high.

With an active area 21 cm wide by 14.5 cm high, the aspect ratio of the actual screen display is 1.45. I can tweak and twiddle the vertical size and horizontal width adjustments the next time I have the monitor apart to change these slightly. When I invoke the CIRCLE command with the JSU I get wider than tall ellipses, but with Minerva or SMSQ/E the circles are very nearly round.

In MODE 4, (pixels across)/(pixels down) for the entire screen is 512/256 giving a pixel aspect ratio of 2.

In MODE 8, (pixel-pairs across)/(pixels down) for the entire screen is 256/256 or a pixel-paired aspect ratio of 1.

Pixel units are used by the commands WINDOW, BLOCK, BORDER, SCROLL, and PAN. Graphics units are used by the commands POINT, LINE, LINE_R, ARC, ARC_R, CIRCLE and CIRCLE_R. The CURSOR command can use both pixel units and graphic units!

Pixels are located with reference to the UPPER left of the screen when locating a WINDOW on the screen. With reference to a particular WINDOW (which may also have a BORDER) the pixel origin for locating BLOCKS in that WINDOW is the UPPER left corner of the active WINDOW inside the BORDER.

Graphic units are relative to the LOWER left corner of a WINDOW, and if a BORDER is used then the graphic origin is the LOWER left corner of the active WINDOW area. The BORDER area is a non-active area.

The graphic aspect ratio for a full screen of 512x256 pixels is about 1.48 for the SMSQ/E and about 1.25 for the JSU.

The graphic aspect ratio for a "pixel square" window such as 200x200 is about 0.74 for the Minerva ROM (which has many 'Screen Tweaks') and also about 0.74 for the SMSQ/E ROM, but is about 0.63 for the JSU ROM.

Pixels to Graphics and vice-versa

So how can we know the graphic units that match the pixel units and the other way around?

Vertical graphic units per pixel

The default SCALE setting (scale_factor = 100, x_offset = 0, y_offset = 0) covers a vertical graphic range from 0 to 100 for a total of 101 'central

graphic values' (remember the fence posts?) with the `x_origin` at 0 and the `y_origin` at 0. These central vertical graphic values are mapped to whatever number of vertical pixels are in the active window. To get a 1-to-1 correspondence the `scale_factor`(in graphic units) should be 1 less than the vertical size(in pixels) of the active window. For the default `scale_factor` of 100 this would be an active window height of 101 pixels. With the maximum active window height of 256 pixels the `scale_factor` needs to be 255 for a 1-to-1 relation to exist between the graphic units and the pixel units. The QL with zero `y_offset` will then match graphic zero to the bottom pixel and match graphic 255 to the top pixel. The graphic vertical range per pixel will be 1 (from $-1/2$ to $+1/2$) graphic unit; the bottom pixel will represent any graphic value in the range $-1/2$ to $+1/2$, the top pixel will represent any value in the range 254.5 to 255.5 graphic units. But what are the horizontal graphic units per pixel?

Horizontal graphic units per pixel

In MODE 4

By experimenting I have found that the range of horizontal graphic values which match to a given pixel(or BLOCK) is a fixed ratio of the vertical graphic range per pixel. This ratio of graphic horizontal range to graphic vertical range I think of as the graphic aspect ratio of a pixel, and I'll use the term `Graspix` for this ratio. The value of `Graspix` depends upon the ROM; for the Minerva(JSL1) or SMSQ/E(HBA) it is about 0.737984775, for the JSU ROM it is 0.626593894 or so. Rational approximations that work well are 476/645 and 344/549 respectively. 16384/22201 and 12973/20704 are better (but not needed?) approximations. I'm curious as to how the QL does it; does anyone know? Note some of the factors for those numbers: $476 = 28 * 17$, $645 = 15 * 43$, $344 = 8 * 43$, $549 = 9 * 61$ while $16384 = 128 * 128$, $22201 = 149 * 149$, $20704 = 32 * 647$, and 12973 is prime. Does the QL ROM intersperse a few bit-shifts, multiplies and divides to figure out where to put the graphic units? Who can (and will) tell?

Horizontal graphic units per pixel-pair

In MODE 8

As you might expect the graphic horizontal range I find per pixel-pair in MODE 8 is twice the range that I find for the pixel in MODE 4. But there is a difference in the location of the 'central graphic value' between the JSU and the other ROMS (HBA and JSL1) that I tried. The JSU puts the central value in the middle (as it were) of

the first (leftmost) pixel of the pair, while the other ROMS put the 'central graphic value' in the middle of the pixel pair!

BORDER

I've not yet taken the `BORDER` command into account; when `BORDER` is used it changes the active window size. The `BORDER` size (in pixels) is applied once at top and bottom, and twice that size at the left and right. Thus the active vertical size of the window in pixel units is 2 times the `BORDER` size smaller, and the active horizontal size of the window in pixel units is 4 times the `BORDER` size smaller. This changes the range of graphic units per pixel (both vertically and horizontally), unless a compensating change is made in the `scale_factor`.

Graspix_bas

I've noodled around with a program "Graspix_bas" that uses `Graspix` for both maximum and random sized windows in MODES 4 and 8 and they look OK so far. Minerva(JSL1) is tolerant and forgiving, but SMSQ/E(HBA) and JSU will balk if the `BLOCKS` are not in bounds. There's a choice of either the `[M]`aximum window size, or a series of `[R]`andomly sized windows. If you choose the random series, you can also choose `[S]`quare' windows. A random `scale_factor` and random graphic origins are used to make a test window, then a sequence of tests are run, first in MODE 4, then in MODE 8:

[1] A maroon `BLOCK` fills the window, and circles centered about the random graphic origin are shown with radii that increase by 10 graphic units.

[2] A grid of white lines should fill every other column and then every other row of pixels on a red background. If the value for `Graspix` is not correct there will appear to be gaps(dark bands) or overlaps(bright bands). Muck about with the `Graspix` parameters to see what happens!

[3] An olive-drab maximum `BLOCK` should fill the active window

[4] Next a pair of diagonal lines to the corners, a `CIRCLE` of maximum size, and an `ELLIPSE` that just fits.

[5] The next event repeatedly puts on random white `POINTS`, followed quickly by a black `BLOCK` meant to cover at the same place on the screen. This checks the conversion from graphic values to pixel values.

[6] A light green `BLOCK` fills the window.

[7] This is followed by repeatedly placing at random a small black `BLOCK`, followed quickly with a white `POINT` meant to cover at the same

place. This checks the conversion from pixel units to graphic units.

[8] A grey BLOCK blots out the window.

The series [1] through [8] is repeated in MODE 8.

[9] A black BLOCK blots out the window; then use [spacebar] to get another window or else use [ESC] to exit.

To keep things moving along, touch [spacebar].

Try it on your system and see what you get! I'd be interested in finding out what the Graspix values might be for the AH, JM, MG and other ROMS.

I'm puzzled by the appearance of some of the circles in test [1]. I'll be noodling with CIRCLE, ELLIPSE and ARC, 'testing the limits' for my next article in this series.

Steve Poole writes from France:

Regarding H.L. Schaaf's request for information on how SCALE is calculated, the graphics scale is:

```
.74*vertical_scale* (window_pixels_across/  
window_pixels_down)
```

■

Letter-Box



J. C. Marcus writes:

Just a quick note with reference to John Wakefield's letter in the latest QL Today. I cannot comment in any depth on his printer problem, as I do not have a colour printer, but there is something about this in Club QL International issue #95, a letter from Graham Lutz. This may help. On machine code tutorials, there is a little 'gem' in the Quanta library. On disk SP05, funnily enough, something called M-CODE TUTORIAL from A. Bridewell. This sets out in a very basic way, how to write assembler code for QL/QDOS. Excellent for anyone wanting to learn about machine code. I hope that helps some.



Dietrich Buder writes:

Concerning some things you wrote in QL Today Vol 2 Issue 4 page 3:

1. Some articles about Z88 will be good. There is a German version of the Z88 and the later Amstrad notepad NC100. I think this is nearly the same as the Z88. I have both in my 'Sinclair Museum'. I had no requirements until today, but I am interested.

2. I use the German word 'QLer' for many years for the English terms 'QL User' or 'QL Users'. I like 'QLers' in English papers too.

3. The text on page 2 of this letter and my file TEXT_UHRCORR_T91 are a shortened form of my text TEXT_UHRKORR_T91 for the 'QL Today Deutsch'. I think the software clock adjust is one of the best basic programs in 1997. The file CLOCKCORR_bas is a part of my BOOT. I wrote lines nnn5 as REMark lines, containing translated English text, please correct as necessary!

My solution for CLOCK ADJUST

In Germany, we have radio controlled clocks (Funkuhr) in the home, on the wrist and in the car, and also for Windows PCs, but not for the QL. Therefore, I asked Stuart Honeyball for a hardware solution to adjust my Gold Card/Super Gold Card with a capacitor. In the end he made a very long and difficult software solution. In QL Today English volume 2 issue 2 I found a short and easy program from Ian Pizer and I fine tuned this solution.

The file from Stuart Honeyball is only for SMSQ/E:

```
d=DATE:d1=d DIV 65536:d2==d MOD 65536  
and later return d=d1*65536+d2
```

QDOS on Minerva or MGG-ROM (German version MG ROM) gives the error message Overflow. QDOS and SMSQ/E work fine (lines 2930, 3010 and 2830) with:

```
d=DATE: h=d/65536:h1=INT(h): h2=h-h1  
and later return d=(h1+h2)*65536
```

h1 and h2 are stored on my file clock_dat (lines 2940 and 3020). Ian Pizer stored only the DATE d with 7 characters. This gives a small error of 17 minutes for the stored time, but his file is easier and works on QDOS too.

If the calculated error is greater than 4 seconds (line 2850) I get a message of the last correction and the calculated error (lines 2870 to 2890) and then the uncorrected clock (line 2910). Now I have time to look on my radio controlled clock. Then comes the corrected clock (line 2260).

I have made the procedure SETCLOCK for the first input or later for changing to summer time or winter time. There is also a short instruction (lines 2990 and 3000).

I have only the command PROT_DATE 1 (line 2270) because the Gold Card or Super Gold Card starts with PROT_DATE 0

Finally, an unknown effect. In line 3020 I have the file name win1_CLOCK_DAT because the filename lq\$&'CLOCK_DAT' doesn't work sometimes.

```

1040 :
1050 lq$='win1_'
2250 :
2260 CLOCKKORR: EW lq$&'Clock_exe': REMark start any clock with EW!
2270 PROT_DATE 1
2780 :
2790 DEFine PROCedure CLOCKKORR
2800 REMark Clock-Corrector (based on Ian Pizer QL, Today Heft 2/1997 Page 38)
2815 REMark start PROCedure SETCLOCK in direct modus after correcting the clock
2820 OPEN #3;lq$&'CLOCK_DAT': INPUT #3;h1,h2,sec: CLOSE #3
2830 d=(h1+h2)*65536
2840 z=DATE-d: k=INT(sec*z/604800)
2850 IF ABS(k)>.4
2860 BEEP 800,60
2870 AT 6,15: PRINT INT(z/3600)!' hours ago the clock has been corrected.'
2880 IF sec>0: a$='forwards': ELSE a$='backwards'
2890 AT 9,15: PRINT 'The clock will be corrected'!ABS(k)!'seconds'!a$;'. '
2900 INK 4: AT 20,40: PRINT 'Press any key!': INK 7
2910 PAUSE: CLS: EW lq$&'Clock_exe'
2920 ADATE +k
2930 d=DATE: h=d/65536: h1=INT(h): h2=h-h1
2940 OPEN #3;lq$&'CLOCK_DAT': PRINT #3;h1\h2\sec: CLOSE #3
2950 END IF
2960 END DEFine CLOCKKORR
2970 :
2980 DEFine PROCedure SETCLOCK
2990 CLS #0: PRINT #0;\,'The clock must be corrected forwards [+] or backwards [-]. '
3000 INPUT #0;,'Please enter the required correction in seconds per week: ';sec
3010 d=DATE: h=d/65536: h1=INT(h): h2=h-h1
3020 OPEN_OVER #3;'win1_CLOCK_DAT': PRINT #3;h1\h2\sec: CLOSE #3: CLS #0
3030 END DEFine SETCLOCK
3040 :

```

Filter Programs

Norman Dunbar

One type of QL program that almost never seems to get a mention in magazines or on disc is the filter program. I can see no reason why these programs are so neglected and hope that the following article helps promote their cause. If not, well, I hope it gives you a thought about different ways of doing things.

A filter program is quite a simple program. It sits between the output of another program and the disc and changes the data in some way from that which was output by the original program just prior to the data being stored on disc.

Filter programs are so called because they sit in between streams of data and filter the data flow through them - just like the filter you use when wine making to remove all the crud before bottling or racking.

For example, you may have a program which sends out line after line of text. There are no headings, no line numbers and no page breaks or page numbers. The program does not put a heading on each (or any) of the (non-existent) pages and is generally not of much use to you.

Of course you may be able to send print to a disc file and then edit it adding line numbers, page breaks, headings etc as required. A filter program can do all of this for you. Read on to find out how.

First of all you must have a command on your QDOS system (I cannot really use the term 'QL' anymore as many of you will be using QXL, QPC, Amigas or STs to do your 'QL' work) that allows a file to be executed and to supply parameters. On Toolkit 2 systems, this will be 'EX' or 'EXEC'. Normal QLs without Toolkit 2 - or something equivalent - will not have this command. If this is the case, then I am sorry but the article will not be of much help to you. Filters are mentioned in chapter 8.2 of the GOLD CARD manual, but I think it is in just about all Toolkit 2 manuals, described under the EX command. Toolkit 2's EX or EXEC commands, Turbo Toolkit's EXECUTE and QLiberator's QX command allow the passing of parameters to a job.

As data passes from one filter to another, there is no change to the original data. For example, a data file is filtered through a program which converts everything to lower case, and then through another to capitalise the first word in every sentence and finally to the printer.

In this example, the original file remains as it is and a couple of transient pipes exist for the duration of the filtering. After the data reaches the printer, it is only the paper copy that exists in the lower case & capitalised version.

Filter programs can be written in just about any language, C, assembler, compiled SuperBasic or uncompiled SuperBasic on the QXL. There is a slight difference if you write compiled SuperBasic filters - Turbo uses channels #14 and #15 as its pipe channels and QLiberator uses #0 and #1. The QXL uncompiled SBasic filters also use channels #0 and #1. C programs use STDIN and STDOUT which usually map onto #0 and #1.

As many filters as you wish can be inserted into a data stream and none of them need to know about any of the others. A well written filter will not care about where its data comes from, it will simply accept the data, filter it in some way and write it out again. The program cares not a jot about where the data comes from or where it is going afterwards.

The general format for executing a filter will be something like:

```
EX program to program to program
```

Where each 'program' is made up of a program name and any parameters that are required by that program, as follows :

```
program_name, input_file, output_file;
"command string"
```

To expand the first example with the details in the second, we could have a command line such as the following :

```
EX prog_1, ram1_input; "1
2 3" to prog_2
```

In this case the executable job called 'prog_1' reads its input from 'ram1_input', requires 3 numeric parameters '1', '2' and '3' and passes its output data to another executable called 'prog_2'.

As you can probably imagine, the command line to execute a large number of filters can be quite complicated. We on the other hand will start off very simple. This first program (Listing 1) takes its input data and converts it to lower case. It is written in C and will be assumed to be compiled to a program called 'ToLower'.

As you can see the filter is very simple. It does not open or close any channels, it does

not seem to do very much, in fact. What has happened is that the filter starts up and finds the input and output channels mapped on to its already opened 'stdin' and 'stdout' channels.

C program always have at least 3 standard channels opened at startup - very similar to #0, #1 and #2 of interpreted SuperBasic programs. In a C program the input is always 'stdin', output is always 'stdout' and errors are always 'stderr'. There may be others for the printer etc but we will not bother with those here.

The above filter reads from its 'stdin' channel, converts any character that is in upper case to lower case and writes it to its 'stdout' channel. Any character that is not in lower case, such as digits, punctuation etc are simply written out unchanged.

You may be wondering where the test is done to check for upper case but in the C library function 'tolower', the character being converted is first tested to make sure that it is in upper case - we don't have to check. As I said - a very simple example.

If you run this program using the command line:

```
EX tolower, input_file, output_file
```

Then all the UPPER case character in input_file will be converted to lower case in output_file.

You can of course pass the input from one filter directly into another. Listing 2 is a variation of the above, and it converts all lower case letters into UPPER case.

Listing 1

```
#include <stdio.h>
#include <ctype.h>

/*-----*
 * TOLOWER - a filter to convert its input to lowercase. *
 *-----*
 * Copyright Norman Dunbar 1997. *
 * Permission given for unlimited use and abuse ! *
 *-----*
 * EX tolower,input,output *
 * (Reads 'input' and writes lower case to 'output') OR *
 * * *
 * EX tolower,input TO program,output_file *
 * (Reads 'input', writes lower case to input of 'program' *
 * and this writes its output to 'output') OR *
 * * *
 * EX tolower,input_file TO program TO program,output_file *
 *-----*/

main()
{
    int ch;

    while((ch = getchar()) != EOF)
        putchar(tolower(ch));
}
```

Listing 2

```
#include <stdio.h>
#include <ctype.h>

/*-----*
 * TOUPPER - a filter to convert its input to uppercase. *
 *-----*
 * Copyright Norman Dunbar 1997. *
 * Permission given for unlimited use and abuse ! *
 *-----*
 * EX toupper,input,output *
 * (Reads 'input' and writes upper case to 'output') OR *
 * * *
 * EX toupper,input TO program,output_file *
 * (Reads 'input', writes upper case to input of 'program' *
 * and this writes its output to 'output') OR *
 * * *
 * EX toupper,input_file TO program TO program,output_file *
 *-----*/

main()
{
    int ch;

    while((ch = getchar()) != EOF)
        putchar(toupper(ch));
}
```

This one converts (only) lower case characters into upper case - just the opposite of the first example. This program will be compiled to a file called 'toupper'.

First, try this :

EX toupper, input_file, output_file

Now when you look at output_file, all the lower case has indeed been converted to upper case. You need a slightly different command line to pass the output from one filter to the input of another. The command is :

EX tolower,input_file TO toupper,output_file

This will take your input_file and pass it in to the program 'tolower' where all its UPPER case characters will be converted into lower case. Then all these characters will be piped into the program 'toupper' and converted back, along with all the original lower case characters to UPPER case and written out to your output_file. The output from the 'tolower'

program is passed along to 'toupper' as it is written out, it is not stored in the QL's memory until it reaches end of file, but simply passed through, just like a filter does. Of course, you could simply miss out the 'tolower' program and just run the first example of 'toupper' to get the same result, but this does show how the 2 filters are linked.

Now things are progressing. In our original example we wanted line numbers as well as other stuff, but we will concentrate on line numbers first. The following filter (Listing 3), again written in C, will read in a file of data and write the same file with line numbers at the start of each line. This one is slightly more complex than the previous

two were, but it is still quite simple.

We have a line_no variable to hold the current line number and a buffer to hold each line as it is

Listing 3

```
#include <stdio.h>

/*-----*
 * LINENO - a filter to convert its input to uppercase. *
 *-----*
 * Copyright Norman Dunbar 1997. *
 * Permission given for unlimited use and abuse ! *
 *-----*
 * EX lineno,input,output *
 * (Reads 'input' and writes to 'output' with line numbers) *
 * * *
 * EX lineno,input TO program,output_file *
 * (Reads 'input', writes with line numbers to input of *
 * 'program' and this writes its output to 'output') OR *
 * * *
 * EX lineno,input_file TO program TO program,output_file *
 *-----*/

#define MAX_LINE 255

main()
{
    int line_no = 1;
    char buffer[MAX_LINE];

    while(!feof(stdin)) {
        fgets(buffer, MAX_LINE, stdin);
        printf("%.5d %s", line_no, buffer);
        line_no++;
    }
}
```

read in by the program. The buffer is at most MAX_LINE (255) characters long (including C's string terminating character) and will not be allowed to exceed this regardless of the length of an incoming line.

The filter uses the C function 'fgets' to read each line of text from the filter's 'stdin' channel and saves it in 'buffer'. Then it prints a 5 digit line number and the contents of the buffer to the output channel, 'stdout'. The last statement simply adds one to the line number, ready for the next line. When end of file is detected, the program simply stops.

So now we have an output file which has line numbers added. How do we get it chopped up into pages? This next example, chops the data stream into pages of a given length.

page and prints the current page number on it. The line number is reset and the page number is incremented.

So we can take our text file, pass it through the LINENO filter to add line numbers and then pass the output from LINENO through this filter, PAGE, to split the text into 65 line pages adding a page number to the top of each new page. This is done with the following command line :

```
EX lineno, input_file TO page, output_file
```

And that is almost all there is to filters. Although I have not shown it, you could write your filter programs so that the starting line number and the increment are passed as parameters.

```
EX lineno, input_file, output_file; '100 10'
```

which would start at line 100 and increment

each line by 10. Of course, the LINENO program must be changed to pick up these parameters before it will work.

You may imagine that filters have to be written in C. This is not the case, only the above examples were written that way. To show that it can be done in SuperBasic, here is the 'tolower' program (Listing 5) rewritten in a manner that is compatible with Qliberator only. Compile with no windows open.

Turbo users can unREMark line 1015 and REMark line 1010 to set the correct channel numbers. Make sure you set the option to copy 0 windows prior to compiling.

Don't forget the semi-colons (;) at lines 1045 and 1055 or you will get a line feed between every character passed through the program!

Notice, however, that we MUST check for upper case characters before we do the conversion. Another problem with this version is that foreign character sets will not

be converted but they will (should) be by the C version. Once again, simply:

```
EX tolower_obj, input_file, output_file
```

To get the whole thing working.

Originally we wanted line numbers, pages and headings on our output. So as I have given you

Listing 4

```
#include <stdio.h>

/*-----*
 * PAGE - a filter to chop the input into 65 line pages. *
 *-----*
 * Copyright Norman Dunbar 1997 *
 * Permission given for unlimited use and abuse ! *
 *-----*
 * EX page, input_file, output_file OR *
 * EX page, input_file TO prog, output_file OR *
 * EX page, input_file TO prog, TO prog TO prog, output_file *
 *-----*/

#define MAX_LINE 255
#define PAGE_SIZE 65
#define FORM_FEED 12

main()
{
    int page_no = 1,
        line_no = 99;

    char buffer[MAX_LINE];

    while(!feof(stdin)) {
        fgets(buffer, MAX_LINE, stdin);

        if (line_no > PAGE_SIZE) {
            printf("%cPAGE %3d\n", FORM_FEED, page_no);
            line_no = 1;
            page_no++;
        }

        printf("%s", buffer);
        line_no++;
    }
}
```

This example keeps a running check on how many lines have been written out by the program and holds a page number. When the number of lines written is greater than the number of lines allowed on a page, PAGE_SIZE, the program forces the printer to take a new

all the information you need to write a filter, why not have a go at changing the PAGE filter so that it prints a heading as well. The heading should be passed in as a parameter using the following command line:

```
EX lineno, input_file TO page, output_file;
'header$'
```

And, if you get that working ok, why not allow the page length to be passed as well. The PAGE example above uses 65 line pages but this may not be suitable for your printer. The command line will now be as follows :

```
EX lineno, input_file TO page, output_file;
'header$ page_length'
Have fun and happy filtering.
```

Listing 5

```
1000 REMark TOLOWER
1005 :
1010 in = 0: out = 1: REMark QLiberator only
1015 REMark in = 14: out = 15: REMark Turbo
only
1020 :
1025 REPEAT lower
1030 IF EOF(#in) THEN EXIT loop: END IF
1035 a = CODE(INKEY$(#in))
1040 IF a > 64 AND a < 91 THEN
1045 PRINT #out, CHR$(a || 32);
1050 ELSE
1055 PRINT #out, CHR$(a);
1060 END IF
1065 END REPEAT lower
```

QL Show USA

Here are the initial details for the 98 North American QL Show in Bedford, PA. Although Bedford is on an interstate exit this is not a heavily populated area. There is no public transportation but the restaurant is within walking distance of the recommended motel and there are several other motels and stores nearby.

Date of the Show: Saturday May 23, 1998

**Location of the Show: Carriage House Restaurant
Exit 11 off the I-70 & I-76 Interstate
Bedford, Pennsylvania USA**

Phone: (814) 623-1174

Bedford is half way between Harrisburg and Pittsburgh.

Time of the show: 9am - 4pm

Format of the show: The show will include talks and demonstrations by well known QL personalities and sales by a number of vendors. The show will take place in the main dining room of the restaurant and lunch is included in admission to the show. After the show a banquet will be held at the same restaurant at 6 PM Saturday evening. All the newest QL hardware and software will be there to see and purchase.

Admission Fees:

\$12 per person if you notify Frank Davis in advance

\$15 per person at the door

This includes admission to the show and LUNCH and general refreshments throughout the day.

Recommended Motel:

Super 8 Motel

Business Rte 220 N

Bedford, PA 15522

Phone: (814) 623-5880 FAX: (814) 623-5880

Also at Exit 11 of the I-70 & I-76 Interstate at Bedford Rates:

Double occupancy with one double bed \$41.29

Double occupancy with 2 separate beds \$43.79

When you make your reservation mention the QL computer show to get this special rate. The rate is per day. There are 57 units, Exercise equipment, HBO, Free local calls, waterbeds, children under 12 free.

Recommended Airports:

DULLES International Airport Washington, DC

(This is about 2 1/2 hours by car from Bedford).

Pittsburg Airport (About 2 hours by car to Bedford).

Harrisburg Airport (About 2 hours by car to Bedford).

There will be a dinner gathering 6 PM Friday night also at the Carriage House Restaurant. Those flying in to airports and needing rides to the show please contact Frank Davis and every attempt will be made to connect you with a local QL person going to the show who can meet you and give you a ride. Likewise, QL people driving to the show who would like to give a ride to a QL enthusiast from far away please contact Frank Davis.

This is the 6th annual North American QL show. It is being sponsored by NESQLUG (The New England Sinclair Users Group) and all details are being handled by:

Frank Davis

FWD Computing

P.O. Box 17

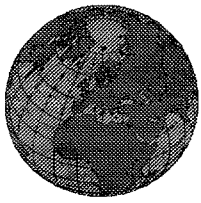
Mexico, IN 46958

USA

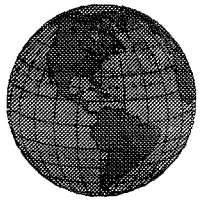
email: fdavis@quest.net

Tel. Tue-Sat, 5-9pm:

765-47308031



The QL Show Agenda



Lots of dates to fill up your Agenda with:

Saturday, 7th of March 1998

QL Show Paris. Unfortunately, this show was reported to us two weeks after the last QL Today was sent out.

Sat./Sun, 28th/29th of March 1998

QL Show in Salzburg, Austria. For details, see German issue of QL Today or contact the organiser Mr. Koll, Tel. 0043 664 160 58 10. **This is an ideal meeting for people from Austria, Switzerland, the southern parts of Germany, Italy, Croatia etc. - of course, everybody is invited!**

Saturday, 18th of April 1998

International QL Show Eindhoven, The Netherlands. The meeting will be held at St. Joris College (same venue as always). We hope this will become one of the big, famous International meetings which were quite successful in the past!

Saturday, 25th of April 1998

East Midlands Quanta Workshop, Selston Parish Hall, Selston, Nottingham, England. Details are shown below.

Saturday, 23rd of May 1998

QL-Show USA. Details on reverse side.

Quanta A.G.M. And Workshop Selston Notts. 25 April

This years AGM/Workshop will be held at Selston Parish Hall, Mansfield Road, Selston. There is ample free parking on site and the venue is close to the M1 (Junct. 27/28). This will be a one day event - but it will be a full day!

Things start at 10:00 with the workshop and talks from traders, in the afternoon will be an open technical forum, so you can put that question that's been bugging you to our panel of experts. This will be followed by the Quanta AGM at 4:30.

Snacks and drinks will be available throughout the day. After the close of the AGM will be a short break followed by a -

Social evening, with buffet and bar!



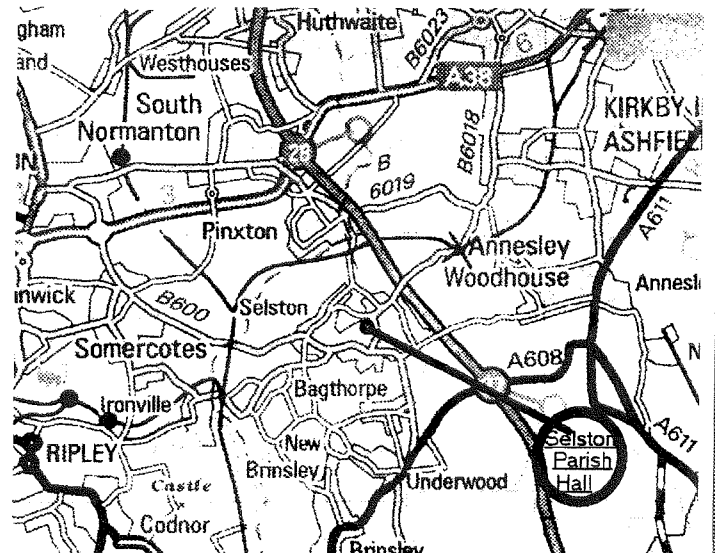
All this is in the same building. Those wanting transport to/from their nearby hotels can make use of a minibus for a nominal charge. The minibus will also take us all back to our hotel after the evening spent in the bar! Things finally come to an end at 11:00PM.



Organised by: - SeQueL, the Notts / Derby subgroup, with funding from Quanta.

There are many nearby attractions for those in your family who have no interest in computers. Why not bring them along to visit Alton towers, Sherwood forest, Chatsworth house, gardens, farm and park, or the American adventure park.

An enjoyable weekend for all the family!!!



Selston is on the B600 Nottingham to Alfreton road.

M1 junction 28 and A38 South Normanton turnoff. Follow the sign for Pinxton (B6019). Follow this road to a 'T' junction, turn right. The Parish Hall is 400 m. on the right.

M1 junction 27. Follow road A608 - Heanor - Underwood - Selston (B600) to the Sandhills Tavern. Turn right and right to the B600 - Alfreton - Selston road. Follow the road until petrol station on the right. Turn right to Mansfield road, B6018 - Kirkby in Ashfield - Sutton in Ashfield. Follow road and Selston Parish Hall is on your left 400 m.

Train to Alfreton Parkway. Bus from Derby to Alfreton 243 or 245, Alfreton to Selston R12 or R13. Bus from Nottingham to Selston R12 or R13.